AFWAL-TR-83-1156

DIGITAL CAPABILITY FOR COCKPIT TELEVISION SENSOR INSTRUMENTATION SYSTEMS

Duane O. Hague, Jr.
Reference Systems Branch
System Avionics Division

November 1983

Final Report for Period 1 August 1980 - 30 June 1983

Approved for public release; distribution unlimited.

AVIONICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Duane O. Hague Jr.

DUANE O. HAGUE, JR.
Project Engineer

Ronald L. Ringo

RONALD L. RINGO, Chief
Reference Systems Branch
System Avionics Division
Avionics Laboratory

FOR THE COMMANDER

Raymond D. Bellem

RAYMOND D. BELLEM, LT COL, USAF
Deputy Chief
System Avionics Division
Avionics Laboratory

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/AAAN, *W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFWAL-TR-83-1156	2. GOVT ACCESSION NO. A139432	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DIGITAL CAPABILITY FOR COCKPIT TELEVISION SENSOR INSTRUMENTATION SYSTEM		5. TYPE OF REPORT & PERIOD COVERED Final Report 1 Aug 80 - 30 Jun 83
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Duane O. Hague, Jr. James E. Barnes		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Avionics Laboratory (AFWAL/AAAN) AF Wright Aeronautical Laboratories, AFSC Wright-Patterson AFB, OH 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Element 62204F, Project 6095, Task 609515, Work Unit 60951528
11. CONTROLLING OFFICE NAME AND ADDRESS Avionics Laboratory (AFWAL/AAAN) AF Wright Aeronautical Laboratories, AFSC Wright-Patterson AFB, OH 45433		12. REPORT DATE November 1983
		13. NUMBER OF PAGES 65
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Digital Encoded Video
Cockpit Television Sensors (CTVS)
MIL STD 1553B
Video Instrumentation
Avionics Instrumentation
Digital Instrumentation
Instrumentation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An approach is presented for incorporating digital capability into a video instrumentation system consisting of a solid-state CCD television camera and video cassette recorder. The approach consists of passive data acquisition from a MIL STD 1553B Multiplex Serial Data Bus with the acquired data being encoded into an unused portion of the video signal. The digital data is encoded as manchester format luminance excursions in the normally blanked horizontal rasters of the video vertical blanking interval. A laboratory

Block 20 (Continued)

prototype of this device (Bus-Monitor/Video-Encoder or BM/VE) was developed along with a device (or Video Decoder) for recovery of the digital data from the video signal. The BM/VE and Video Decoder were used in laboratory testing to establish the performance parameters and design requirements needed for an actual airborne application of the BM/VE approach. A video encoding format for fault-tolerant data recovery is also developed and presented. Test results demonstrate that the primary limitation of this approach is the video cassette recorder bandwidth and that even the lower bandwidth video recorders can support application-dependent capacities of 1-3 thousand digital 16-bit words per second.

FOREWORD

This report describes an in-house effort conducted by personnel of the Reference Systems Branch (AFWAL/AAAN), System Avionics Division (AFWAL/AAA), Avionics Laboratory, Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio. The effort was performed under Project 6095, "Inertial Reference and Guidance Technology," Task 609515, "Reference Systems Algorithm Development and Evaluation," Work Unit 60951528, "Development of a MIL STD 1553 Passive Bus-Monitor/Video-Encoder."

The work reported herein was performed during the period 1 August 1980 to 30 June 1983 under the direction of the author, Duane O. Hague Jr. (AFWAL/AAAN-2), project engineer. The report was released by the author in November 1983.

The author wishes to thank Mr. James Barnes (AFWAL/AAAN-2) for the report section on Development Software and for the development of the complex real-time software that allowed this effort to be successful. SSgt Bernard Nagle (AFWAL/AAAN-2) also made major contributions to this effort by his timely and accurate in-house construction of complex electronics assemblies.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ABBREVIATIONS

1553B	MIL STD 1553B Multiplex Serial Data Bus
BM/VE	Bus-Monitor/Video-Encoder
CPU	Central Processing Unit
CTVS	Cockpit Television Sensor
DMA	Direct Memory Access Input/Output
EPROM	Electrically Programmable Read Only Memory
FIFO	First-In-First-Out Buffer Stack
GEN-Lock	Internal Video Generator Lock Synchronization to External Video Source
KOPS	Thousand Operations (Instructions) Per Second
KWORD	1024 Words (16 bits per word)
LSB	Least Significant Bit
MSB	Most Significant Bit
NRZ	Non-Return-to-Zero serial data format
NTSC	National Television Standard Color, Modification of RS-170
PCIO	Direct CPU Program Controlled Input/Output
RAM	Random Access Memory
ROM	Read-Only Memory
RS-170	ANSI Standard RS-170 for Black/White Composite Video
TTL	Transistor-Transistor-Logic, +5 Volt DC Power
VCR	Video Cassette Recorder
VTBC	Video Time-Base Corrector

SECTION I

INTRODUCTION

1. BACKGROUND

During the period of 1978-80, we became involved in the in-house development of a minicomputer system that provided primary ground-support data reduction for two laboratory flight test programs of gunnery fire-control systems. These programs were the Director Evaluation Feasibility Test, Phase I (DEFT-I) and the All-Aspect Gunsight Evaluator For F-15 (AAGE-15). Both programs used Heads-Up Display systems and Cockpit Television Sensor (CTVS) instrumentation systems consisting of solid-state Charge-Coupled-Device cameras and video cassette recorders (VCRs). DEFT-I and AAGE-15 both used special circuitry to luminance-encode digital data into the unused horizontal portion of the vertical blanking interval of the video from the CTVS camera. The digital data was thus effectively multiplexed into an unused portion of the VCR recording bandwidth. This digital data was encoded at a bit rate of 1.2 megabits/second with a maximum data recording capability of 1400 16-bit words/second with a data sampling rate fixed at 30 hertz. The ground-support minicomputer system that we developed proved capable of recovering this digital data from the video cassettes (recorded in-flight) with typical recovery rates of 97.8% to 99.6%. The primary cause of failure to recover particular digital data regions on the video cassette was aircraft acceleration affecting the tape-tensioning mechanism of the VCR during recording. Additional problems were identified in the areas of horizontal video instability, error detection/recovery, data formatting, and timing conflicts between the video recording rate and the data acquisition rate. However, based on our experience, none of these problems was fatal to this basic approach of adding digital recording capability to a CTVS instrumentation system. In any airborne instrumentation application requiring the recording of both image data and digital data, the ability to use one recorder for both data types could provide several advantages. These advantages are reduced instrumentation volume and the synchronization of digital and image data. The electronics necessary to multiplex

the digital and image data for CTVS instrumentation could easily achieve volume/cost/reliability advantages over the alternate approach of separate digital and video instrumentation systems.

2. BASIC CONCEPT

The basic concept of this project developed from three factors during a series of discussions between Mr. Hague and Maj Edward Jessup (then of AFWAL/AART-1) between May and July of 1980. The first factor was the success of multiplexing digital data into a CTVS instrumentation system without loss of image data for the DEFT-I and AAGE-15 flight test programs. The second factor was the Aeronautical Systems Division (ASD) plan to use CTVS technology in the replacement of the traditional fighter aircraft gun camera. The third factor was the increasing use of the MIL STD 1553B Multiplex Serial Data Bus as the communications link between avionics systems and/or subsystems. The concept resulting from these discussions was that of a "Black-Box" electronics assembly that would provide digital recording capability for the CTVS instrumentation systems designed for airborne applications. This "Black-Box" would have two functions. The first function would be to "monitor" the MIL STD 1553B Multiplex Bus for digital instrumentation data, and the second function is to multiplex (or "video-encode") the acquired digital data into the video signal from the CTVS camera for recording on the CTVS VCR. This "Black-Box" (or MIL STD 1553B Bus-Monitor/Video-Encoder) acts as the instrumentation "bridge" between the CTVS and MIL STD 1553B technologies. The Bus-Monitor/Video-Encoder (BM/VE) concept has three special attributes. The first attribute is passive acquisition of all message traffic on the 1553B Multiplex Bus which would remove instrumentation software task loading from the real-time avionics equipment. The second attribute is programmable hardware/software message selection based on the device address field embedded within the messages (and/or message types) with message-acquisition time-tagging. This attribute is critical since only about 5% of the video signal can be used for encoding digital data without loss of image data. Also, since the 1553B Bus operates in an asynchronously polled mode, typically 30-50% of the message traffic is non-data overhead. The third attribute is the synchronization, formatting, and luminance encoding of the selected data messages for

multiplex insertion into the unused horizontal scan lines of the video signal.

A possible application of the BM/VE concept is given in Figure 1. The digital data acquired from the 1553B-Bus might include flight-profile data, various target sensor tracking data, and weapons control data such as gun-trigger flags. The video cassette would then be removed from the aircraft for post-flight processing on an Aerospace Ground Equipment computer support system. The functions performed by the support system could include recovery, quick-look display, and archival storage of the digital data. Another function might be the display of the video image data overlaid with alphanumeric/symbol status information for evaluation purposes during pilot debriefing. Since the digital data acquisition function of the BM/VE concept is fully programmable, a standardized hardware version of the BM/VE concept could be adapted to a wide range of digital instrumentation requirements. This adaptation could be as simple as plugging in an "application-pack" of Read-Only-Memory containing data-acquisition software parameters. The BM/VE concept in conjunction with CTVS systems can meet digital/video instrumentation requirements in flight-testing, dry-fire gunnery scoring, combat/mission documentation, operational training, and the in-flight collection of maintenance data. In our opinion, a production version of the BM/VE could be developed with a volume of about 0.2 cubic feet and a unit cost of less than $10,000. Given the pre-existence of a CTVS instrumentation system, the BM/VE concept can meet digital recording requirements for a wide range of applications as a cost/space-effective alternative to separate digital instrumentation systems.

3. OBJECTIVES AND SCOPE OF EFFORT

We viewed the BM/VE concept as offering a significant improvement in avionics instrumentation. As a result, an in-house work unit to establish the parameters of the BM/VE concept was proposed and approved in August, 1980. The goal of this work unit (Project 60951528) was to develop a laboratory prototype of the BM/VE. This laboratory prototype

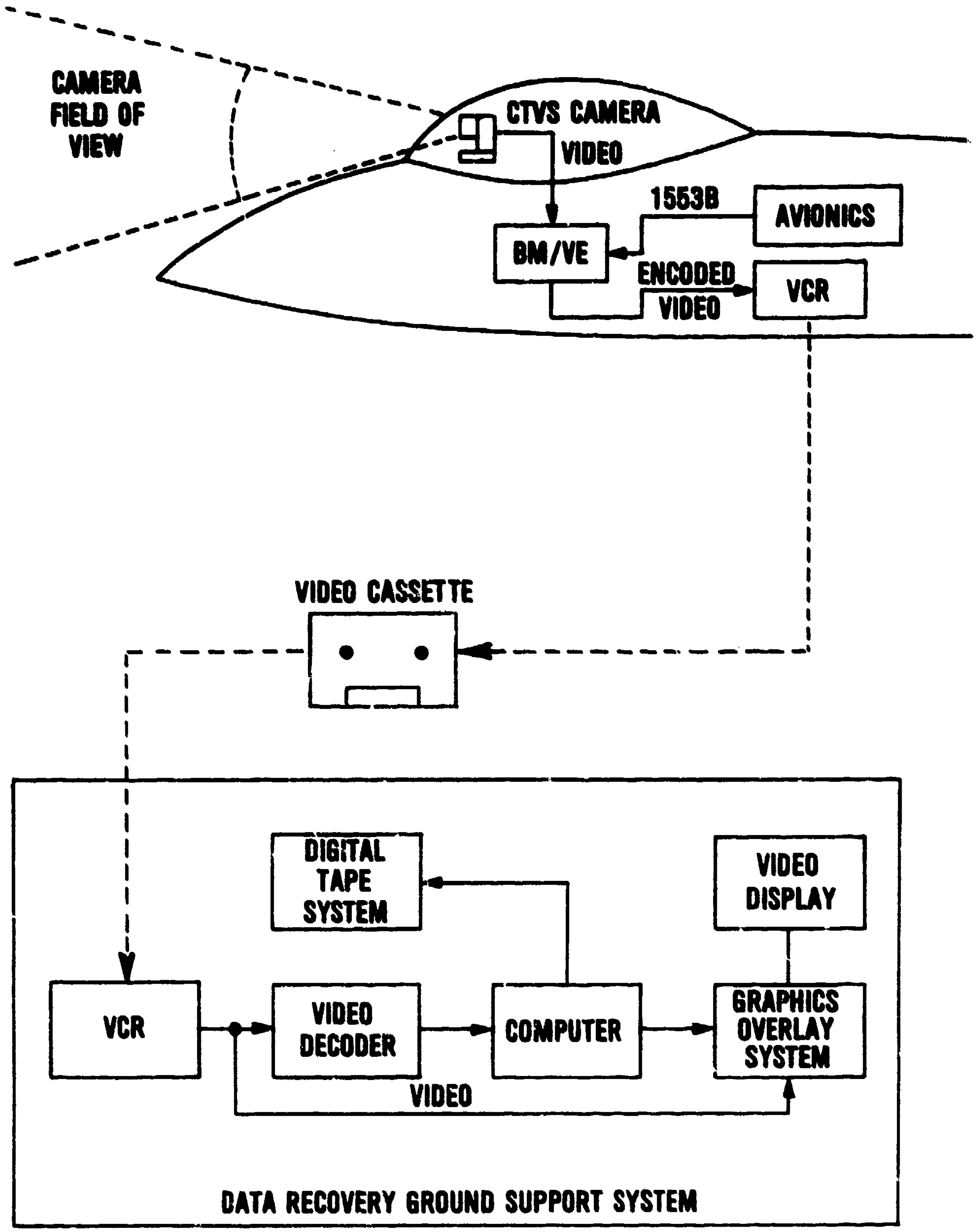


Figure 1. BM/VE Concept Application

would be used to explore and test the design variables of the BM/VE concept. The specific technical objectives for this work unit were to:

- Determine the optimum encoding format versus data capacity
- Determine requirements and procedures for programmable passive-data-acquisition
- Determine the minimum requirements for data recovery
- Establish baseline performance
- Determine the minimum microprocessor requirements
- Develop fault-tolerant data-handling procedures

Achievement of these technical objectives required two support development tasks besides development of the prototype BM/VE. The first support development was a device to provide a valid simulation of a MIL STD 1553B Multiplex Bus including fault simulation. The second support development was a "Video Decoder" device that would recover the digital data from the video signal. Successful results from this work unit would provide the technical baseline for any decision to apply the BM/VE concept to real-world applications in avionics instrumentation.

4. WORK UNIT HISTORY

The development chronology of this work unit is given below:

AUG-SEP 80:	Work Unit "start-up"
OCT-DEC 80:	Preliminary system design and procurements
JAN-APR 81:	Development and testing of basic microprocessor system
FEB 81:	Start of major redesign, timing simulations determine that the Bus-Monitor function requires embedded microprocessor support. Problem resolution results in 150% increase in Bus-Monitor circuitry.
APR-SEP 81:	Design, build-up, and development of slave microprocessor system for embedding in Bus-Monitor
OCT-DEC 81:	Design of 1553B Bus Simulator. Redesign of Bus-Monitor Decoder Card due to software timing conflict

JAN-APR 82:	Build-up and development of final hardware for 1553B Bus Simulator and Bus-Monitor Assembly
MAY-JUL 82:	Data-acquisition-loop testing of Bus-Monitor function with all design goals achieved
JAN-NOV 82:	Design, buildup, and development of Video-Encoder function
SEP-NOV 82:	Major technical problems with holding slave synchronization to all types of RS-170 and NTSC composite video. Problems were resolved by redesign based on experimental data. The Video-Encoder function achieved all design goals.
NOV-DEC 82:	Real-time software debug and testing. BM/VE achieved all design goals for real-time data acquisition and video encoding.
NOV 82-JAN 83:	Design of Video Decoder Assembly and build-up starting in January. Support hardware for BM/VE testing.
JAN-APR 83:	Video Decoder Assembly build-up with schedule delays due to technician availability.
MAY-JUN 83:	Final BM/VE testing with Video Decoder. Determination of data recovery/reliability at various video encoding frequencies.

SECTION II

LABORATORY DEVELOPMENT SYSTEM

This section deals with the prototype hardware developed for laboratory testing of the BM/VE concept. The explanations for the theory of operation of the BM/VE are based on functional block diagrams. The same general approach will also be used in Section III. The primary reason for this approach is that this report is not intended to be an application hardware specification of the BM/VE concept. This report is intended to provide the functional requirments for the BM/VE concept and to identify design approaches and problem areas. For example, many of the programmable BM/VE features (such as encoding frequency) would be "hard-wired" in actual applications. This "hard-wiring" would decrease BM/VE complexity by about 25% over the laboratory prototype. Also more use of large and very-large scale integration circuit technology could be made by using the results of this project which could result in a further 25% decrease in complexity. As a final factor, inclusion of design details would result in an oversize report. There are about 375 square feet of BM/VE engineering drawings. The laboratory system contains about 1000 integrated circuits (primarily small and medium scale integration) of BM/VE custom interfacing and about 500 integrated circuits of test-support electronics. It should also be noted that, while the laboratory system is fully operational, the detailed design includes development compromises resulting from form-fit and parts-availability problems. The details of hardware/software design can be requested by qualified organizations by contacting the office given in Block 9 of the DD Form 1473.

Since the laboratory prototype system contains commercial components of hardware/software as well as the components developed in-house, it must be recognized the names of these commercial components are registered trademarks. The registered trademarks of the Digital Equipment Corporation (DEC) used in this report are:

PDP, RSX, UNIBUS, QBUS, LSI-11

1. BUS-MONITOR/VIDEO-ENCODER SYSTEM

The block diagram of the laboratory prototype BM/VE system is given in Figure 2. A Digital Equipment Corporation LSI-11/23 Microprocessor is used as the main Central Processing Unit (CPU) of the BM/VE. The LSI-11/23 is the direct controller of the Video-Encoder Interface Assembly and the indirect controller of the Bus-Monitor Interface Assembly. Direct control of the data acquisition function is accomplished by a slave CPU embedded into the Bus-Monitor Interface. This slave or Monitor CPU is an Intel 8086 Microprocessor. The Main CPU has over-ride mode-control and access to the Monitor CPU and the Monitor Bus. The Main and Monitor CPUs have program interrupt cross-linkage and also share a memory common via a Dual-Port RAM/PROM (Random-Access-Memory/Programmable-Read-Only-Memory) Interface. The Main CPU uses this Dual-Port RAM/PROM to provide the Monitor CPU with task programs and to access the acquired data. The Main CPU also has an on-line PROM Programmer Interface for permanent entry of Monitor CPU subroutine programs. The program tasks performed by the Main CPU are:

a. Gaining and holding synchronization with the external video source via the video locking and control functions of the Video-Encoder Interface.

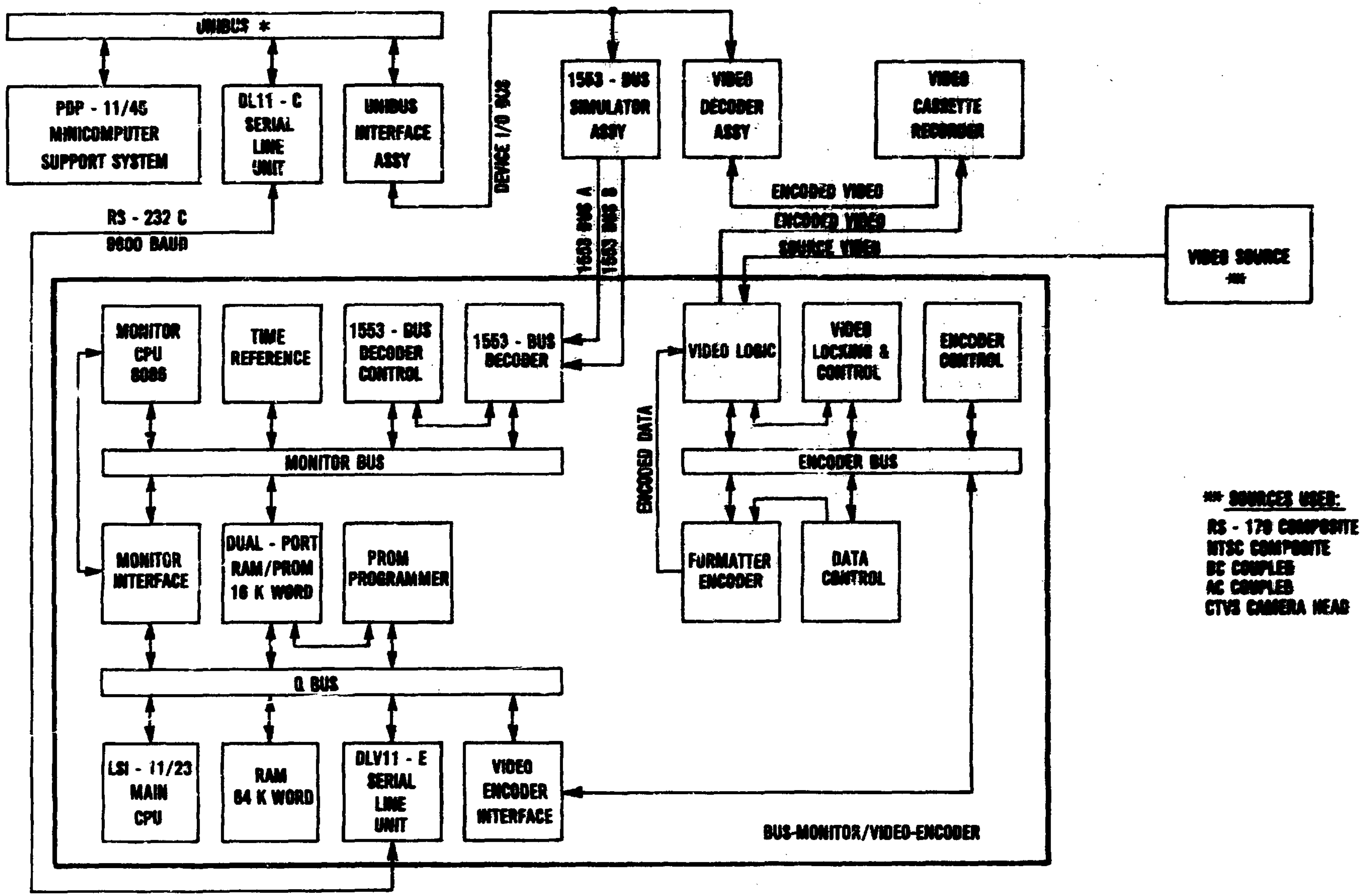
b. Oversight of the Bus-Monitor operations and the interchange of empty and full memory data buffers for acquired 1553B message data blocks.

c. Synchronization and queuing of the full 1553B message data blocks to the encoding function of the Video-Encoder Interface.

d. Exception handling and error recovery for the Bus-Monitor and Video-Encoder Interfaces.

Execution of tasks a through c must be real-time. Task d may or may not be real-time depending on what exception or error occurs, but it must restore real-time operation.

The Bus-Monitor subassembly performs the actual data acquisition from a dual-channel 1553B Multiplex Bus under direct control of the



* TRADEMARK DIGITAL EQUIPMENT CORPORATION

FIGURE 2. BM/VE SYSTEM BLOCK DIAGRAM

Monitor CPU. The Decoder and Decoder-Control Interfaces receive all 1553B traffic and perform hardware message grouping of the serial data word transmission sequence. It should be noted that we used the Instrumentation Bit Option of MIL STD 1553B (Para. 4.3.3.5.3.4) which is also used in the F-16 applications. This option allows simpler hardware message grouping and message-grouping-error recovery within two messages. Without this option, hardware message grouping is slightly more complex and error recovery could not be made until a 30 microsecond gap in serial activity was detected after error occurrence. As each serial message is received, the Terminal Identification Field from the message-control-word is tested against a programmable look-up table to identify message treatment. If enabled, the entire message is loaded into a First-In-First-Out (FIFO) buffer and the Monitor CPU is notified of an end-of-message. The FIFO buffer can hold at least 1.6 milliseconds of 1553B activity and allows averaging of the Monitor CPU load. Based on end-of-message, the Monitor CPU deletes error-free non-data messages, time-tags data messages, and assembles data messages into the message block buffers supplied by the Main CPU.

The Video-Encoder subassembly performs two inter-related functions under direct control of the Main CPU. The first function is to synchronize (or GEN-Lock) the internal video timing generator to the external video source. The GEN-Locked video timing generator provides the timing references for all Video-Encoder activity. The second function is the generation and insertion of the encoded-data signal into the source video for output from the BM/VE. This encoded-data signal includes hardware generated formatting information as well as 1553B data words from the Bus-Monitor Interface.

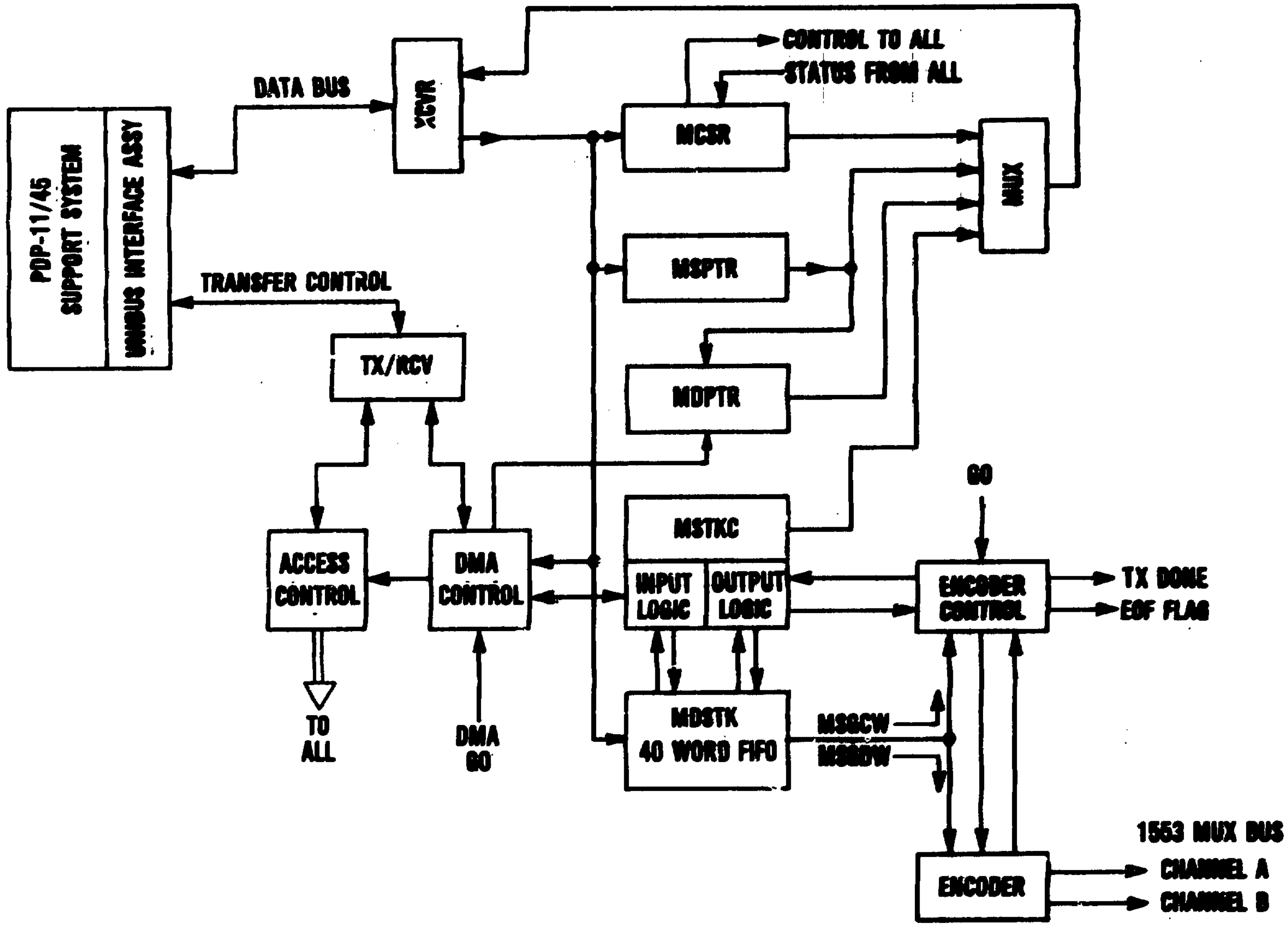
Operation of the BM/VE prototype is supported by a Digital Equipment Corporation PDP-11/45 Minicomputer. The PDP-11/45 provides three separate support functions to the BM/VE. In the first function, it emulates a "Console Terminal" to the LSI-11/23 to provide down-line program loading and control/status communications via an RS-232 serial data link. The second function is the real-time control of the 1553-Bus Simulator Assembly which includes providing the control-data files that result in the dual-channel 1553 Multiplex Bus input to the Bus-Monitor Interface.

The third function is real-time control of the Video-Decoder Assembly which is used for testing the data recovery/reliability parameters of the BM/VE.

2. MIL STD 1553B BUS SIMULATOR ASSEMBLY

A block diagram of the Bus Simulator is shown in Figure 3. This device generates a simulated 1553B Bus profile by sequentially processing a data file of word pairs provided by the support system. Each word pair contains a Message Data Word (MSGDW) and a Message Control Word (MSGCW). MSGDW provides the encoder with the data bits for actual transmission. MSGCW provides the encoder control with the transmission parameters. These parameters include channel select, transmission type, the delay to next transmission (0-4 milliseconds), control bits for forcing transmission errors, and End-Of-File flag. The three transmission modes of the Bus Simulator are the Program Control Mode, the DMA Mode, and the DMA Loop Mode. In the Program Control Mode, the support minicomputer preloads the Message Data Stack (MDSTK) with up to 20 MSGDW/MSGCW word pairs and issues a GO command. The contents of MDSTK are then transmitted over the 1553B Bus. In the DMA Mode, MDSTK is incrementally loaded from a Start-Of-File Pointer via direct memory access to main memory until the End-Of-File flag is detected in a MSGCW data word. The device-busy then terminates after transmission of the word pair containing the End-Of-File. In the DMA Loop Mode, when the DMA logic detects End-Of-File, the DMA Pointer is reset to the Start-Of-File Pointer. In the Loop Mode, the data file transmission is automatically repeated until the support minicomputer exercises over-ride control.

Two support software programs were developed for the Bus Simulator. The first program generates the message data files based on operator inputs of the desired 1553B Bus profile. The second program provides the device driver software and the real-time control and message transmission scheduling required to achieve accurate 1553B Bus simulation. Since all transmission parameters were programmable, worst case and typical case profiles could be generated for burst-data-rate, average-data-rate, message types, 1553B terminal count, and transmission errors.



DEVICE REGISTERS	FUNCTION
MCSR	MESSAGE CONTROL STATUS
MSPTR	MESSAGE START OF FILE POINTER
MDPTR	MESSAGE DMA POINTER (READ-ONLY)
MSTKC	MESSAGE STACK WORD COUNTER (READ-ONLY)
MDSTK	MESSAGE DATA STACK (WRITE-ONLY)

FIGURE 3. MIL STD 1553B BUS SIMULATOR BLOCK DIAGRAM

3. BUS-MONITOR INTERFACE ASSEMBLY

The Bus-Monitor Interface consists of the Microprocessor Section (block diagram shown in Figure 4) and the Decoder Section (block diagram shown in Figure 5). From the hardware point-of-view, the Bus-Monitor is an independent CPU system once it has been initialized by the LSI-11. While the LSI-11 does have over-ride control of the 8086 Monitor CPU, program task synchronization between the LSI-11 and 8086 is normally a software function achieved through use of program interrupt cross-linkage and/or the use of flag words in common memory. The 8086 communicates with the other Bus-Monitor components over the Monitor bus structure which is very similar to the QBUS (except for the interrupt structure). To the LSI-11, the Bus-Monitor appears to be a contiguous block of 256 registers located in the "Bank 7" Peripheral Device address region. The lower 254 register addresses form a window into the 8086 I/O device address range while the other two register addresses are used as an 8086 control/status register and an interrupt linkage control register. The control/status register provides reset/go control and the ability for the LSI-11 to take over-ride access control of the 8086 I/O page. The control/status register is also the source of two maskable interrupts to the LSI-11, one interrupt for 8086 stopped and one interrupt for an access error (either 8086 Monitor Bus Time-Out or improper LSI-11 access of the 8086 I/O page). The interrupt linkage register allows the LSI-11 to assert four maskable interrupts and one non-maskable interrupt to the 8086. The interrupt register is also the source of two maskable interrupts to the LSI-11 that are asserted by 8086 software. This interrupt cross-linkage forms the basis of real-time synchronization between the LSI-11 and 8086.

The Bus-Monitor Microprocessor Card contains the 8086 CPU, the data path logic, and the 8086 interrupt logic. The interrupt logic provides vectored interrupts to the 8086 based on four request lines from the LSI-11 and four request lines (one spare) from the Monitor Bus. A fault vector is also used to provide for the case of LSI-11 actions clearing the request before the 8086 can respond to the request. This logic also "ORs" the Non-maskable Interrupt from the LSI-11 (normally used only for over-ride purposes) with a Monitor Bus Access-Time-Out error condition.

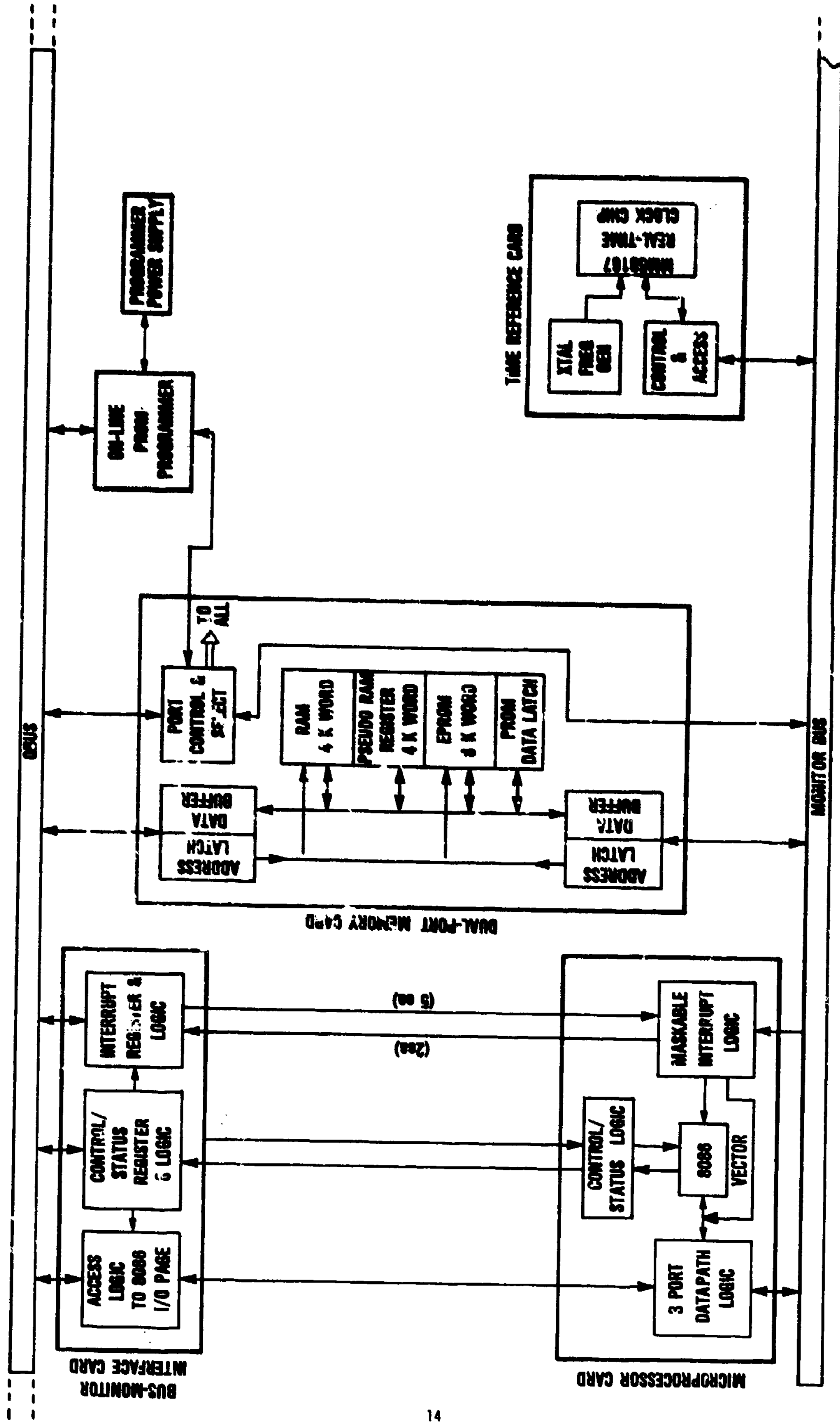


FIGURE 4. BUS-MONITOR BLOCK DIAGRAM, MICROPROCESSOR SECTION

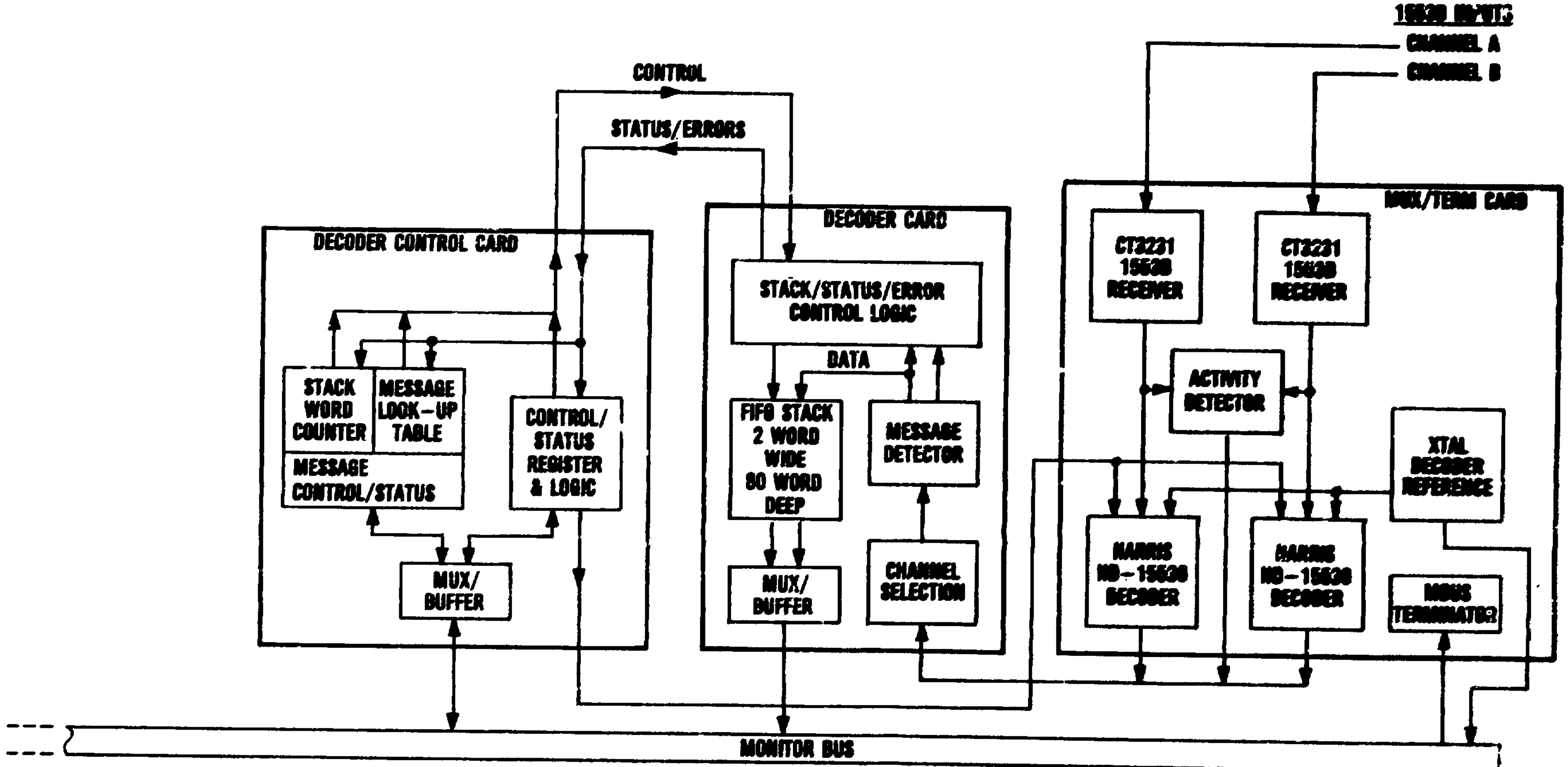


FIGURE 5. BUS-MONITOR INTERFACE ASSEMBLY BLOCK DIAGRAM, DECODER SECTION

The interrupt logic also provides a 2-bit write-only register at the top of 8086 I/O space that allows the 8086 to set either of two interrupt requests to the LSI-11.

The second communication channel between the LSI-11 and the 8086 is a 16K Word Dual-Port Memory Card. This card contains 4K Words of RAM, 4K Words of Pseudo RAM (a 16-Bit register that responds to a 4K address range), and 8K Words of EPROM. This memory is normally initialized with the 8086 program as part of the LSI-11 bootstrap operation. The RAM is normally used for the 8086 vector table, an 8086 task scheduler, and as data common. As various data acquisition tasks were developed and debugged in the RAM, these tasks would then be relocated into the EPROM memory via use of the On-Line PROM Programmer.

The Time Reference Card uses the National Semiconductor MM58167 Real-Time Clock to provide millisecond accuracy time-tagging of acquired 1553B data. This time reference can be "slaved" to any IRIG time data that appears on the 1553B Bus. This card can also provide a time-interval interrupt to the 8086.

The Decoder Section performs the actual data acquisition under control of the 8086 program. This section consists of the Decoder Control Card, the Decoder Card, and the MUX/TERM Card. The MUX/TERM Card receives inputs from two 1553B channels and converts each data channel into serial NRZ data signals and associated status information. This card also provides channel activity detectors that allow the Decoder Card to perform channel selection. The activity detectors operate on the input to the Harris 1553 decoder chips which have an input-to-output delay of about 5 microseconds. This delay allows the Decoder Card channel selection logic enough time to operate. The MUX-TERM Card also provides resistive termination for the Monitor Bus as well as various crystal clock references for the Bus-Monitor Interface Assembly.

The Decoder Control Card is a two-register device that controls the Decoder Card to achieve the Bus-Monitor functions. These controls include Decoder-enable, individual channel enable/resets, and a 32-bit look-up table that provides individual enable/disables for FIFO data stack entry of messages based on the Terminal ID number of a message. The Decoder

Control Card also provides status signals such as "loaded-wor ount" for the FIFO data stack, channel A/B raw activity, enabled-message-busy (receipt in progress), and fatal errors that clear the Decoder-enable. These fatal errors are Monitor Time-out (i.e., 100 milliseconds without receipt of an enabled message), simultaneous activity on both 1553 channels, and data stack overflow. This card also provides an error interrupt and an "end of enabled message" interrupt to the 8086.

The Decoder Card performs the actual data acquisition/selection. Inputs from Channel A or Channel B are selected on a first-come-first-served basis. Once a channel is selected, any activity on the other channel is treated as a fatal error that terminates monitoring operations pending a software restart. The selected channel activity is then grouped into messages based on logical detection of start-of-message and end-of-message. Start-of-message is defined as the first word in an activity envelope having the command/status sync preamble waveform with the Instrumentation Bit=1 (F-16 Option). If the F-16 option were not implemented, monitor start-up or error resynchronization would require detection of a valid end-of-message before a valid start-of-message could be declared. This would result in "throwing away" at least one good message on error recovery and would require more complex logic. As we were not attempting to "prove" the ability to monitor a 1553 Bus, we chose to implement the Instrumentation Bit Option (MIL STD 1553B, Para. 4.3.3.5.3.4). At start-of-message the terminal address field (5-bits) of the received word is latched as a group message identifier. This group identifier is used to select an enable/disable bit from the Message Look-Up Table (Decoder Control Card). If enabled, all received words for the message are loaded into the FIFO data stack for recovery by the 8086. This stack is two words wide and 80 words deep. Each enabled-received word is loaded into the stack at the same time as a decoding-status word. The 8086 removes data from the stack in word pairs of decoding status and then decoded data (which automatically pops the stack). This decoding status word contains the group message ID and flags for start-of-message, end-of-message, channel source, word preamble sync type, parity error, and message format error. End-of-message is based on receiving a 1553 status word (i.e., command/status preamble and Instrumentation Bit=0) with a

terminal ID matching the group ID combined with termination of channel activity. End-of-message will also be forced by detection of a message format error. A message format error is declared when a start-of-message is followed by:

- A parity error in any command/status work
- Detection of second start-of-message
- Detection of a channel change within the message
- Detection of 24 microseconds of channel inactivity

The oc;urrence of a message format error loads a pseudo-word-pair (only error bits valid) into the stack and the message detector returns to looking for a start-of-message.

4. VIDEO-ENCODER INTERFACE ASSEMBLY

The Video-Encoder Interface consists of the Video Section (block diagram shown in Figure 6) and the Encoder Section (block diagram shown in Figure 7). Due to the growth of the Bus-Monitor, it was necessary to construct the Video-Encoder in an expansion card-file with ribbon cables to a general purpose QBus Interface Card in the main BM/VE chassis. This QBus Interface Card supports DMA and up to 4 interrupt sources and 16 I/O device registers. In addition to the QBus Interface, the Video-Encoder has a direct-coupled video input for the source video signal (RS-170/NTSC 1 volt composite terminated to 75 ohms) and a direct-coupled video output for the video with encoded data (RS-170/NTSC 2 volt composite unterminated, 1 volt composite terminated to 75 ohms). The encoded data consists of a block of contiguous video horizontal line rasters in each video field containing words encoded as back-white luminance excursions in a manchester format. Each horizontal line raster has an internal structure of a command word cell followed by 0-7 data word cells. The command word cell contains status flags, the number of data word cells to follow on the line, and a "line-of-block" identifier. The encoded data block has an additional format structure (overlaid by hardware) in that the first and last raster lines of each encoded block contain two format words with the only difference between these raster lines being

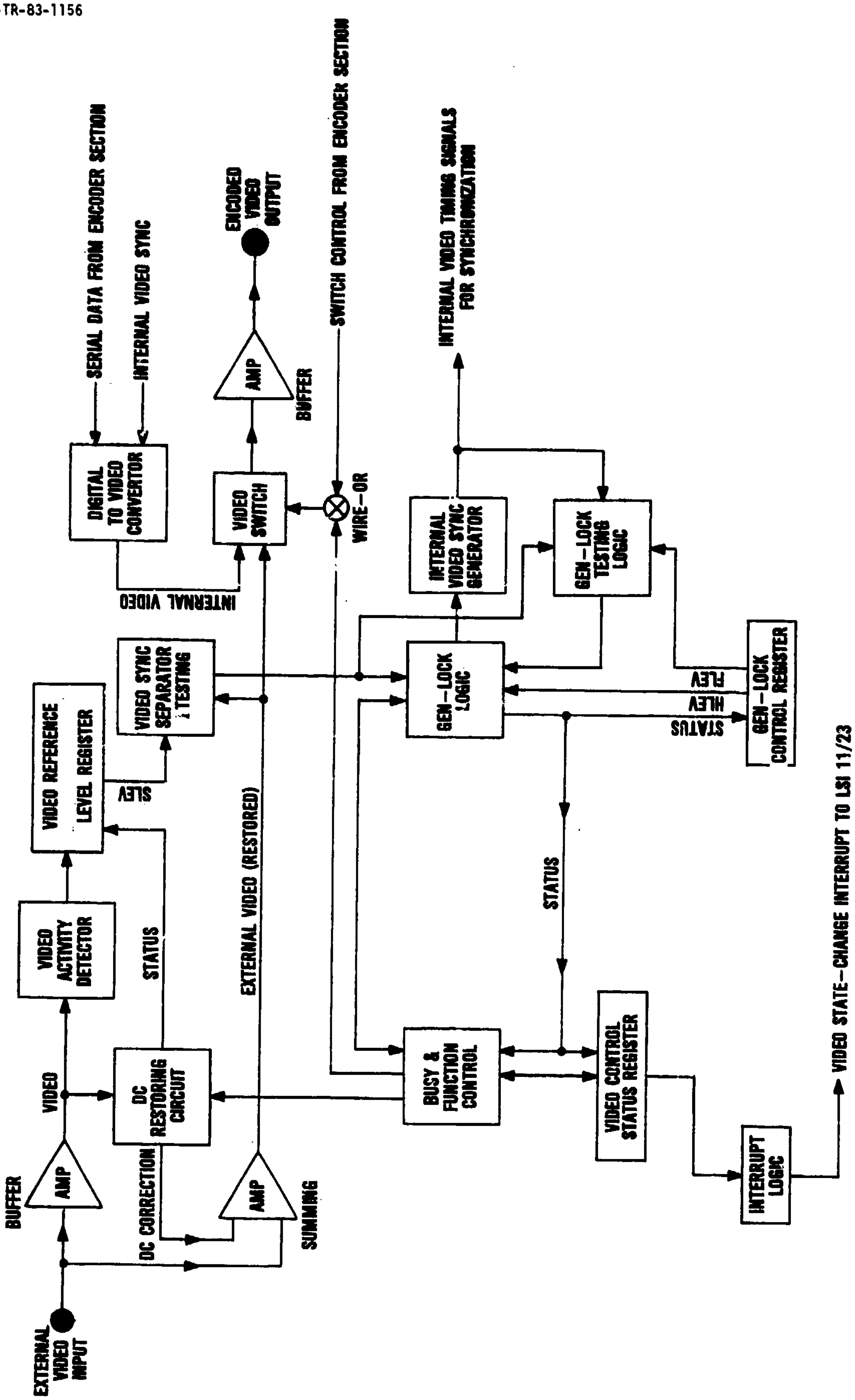


FIGURE 6. VIDEO-ENCODER BLOCK DIAGRAM, VIDEO SECTION

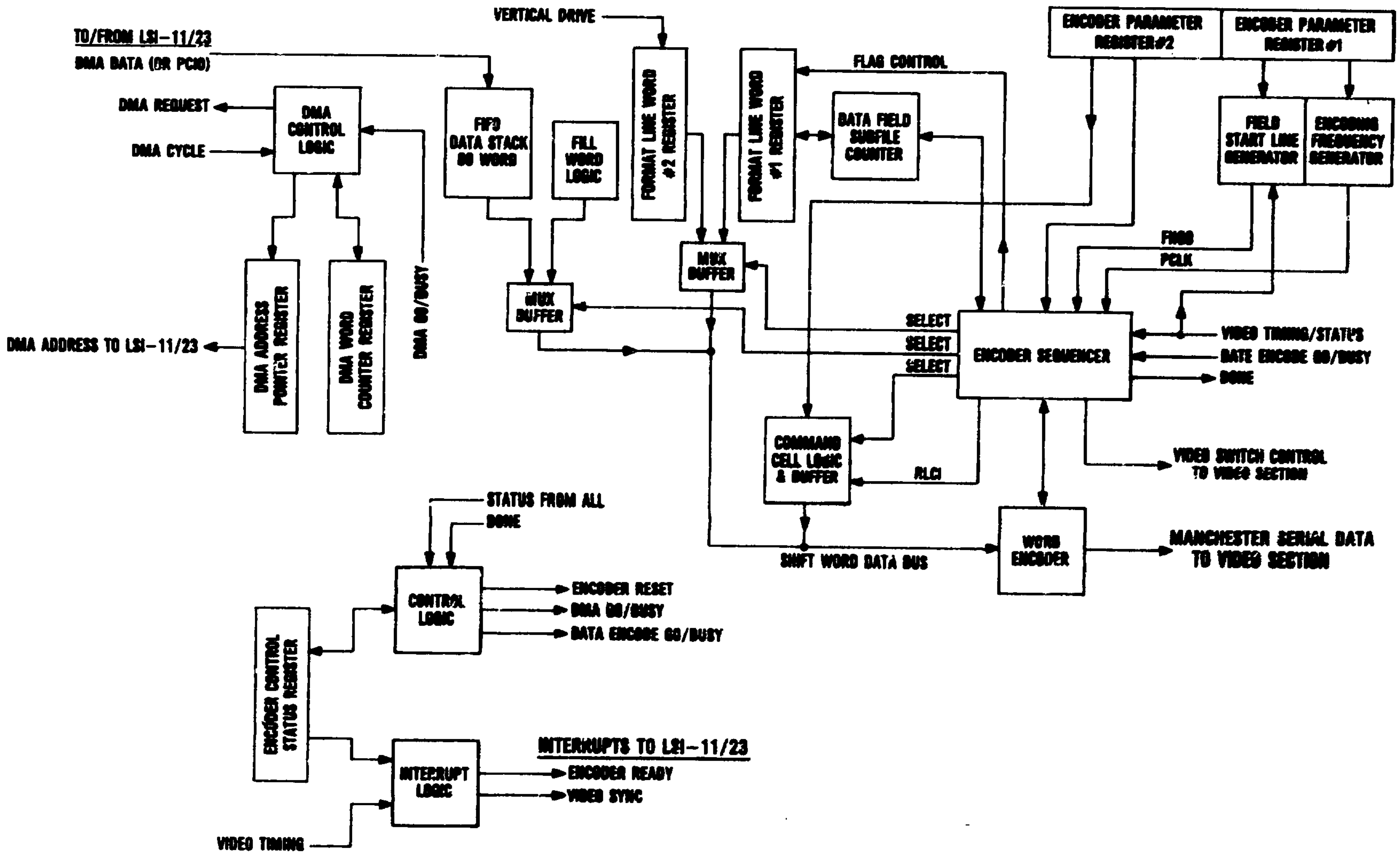


FIGURE 7. VIDEO-ENCODER BLOCK DIAGRAM, ENCODER SECTION

the "line-of-block" identifier in the command cell word. The format words provide a video time reference of a 20-bit video-field counter (i.e., 60 hertz) and provisions for defining a "data-file" of 1 to 16 sequential field encoding-data-blocks. The development of this encoding format is discussed in detail in Section III.

To the LSI-11/23, the Video-Encoder is a set of 11 read/write registers and the source of 3 maskable program interrupts. Individual register bits may be read/write, read-only, write-only (read zero), read/conditional-write, or conditional-write-only. All operations of the Video-Encoder are the direct result of LSI-11/23 operations. The LSI-11/23 performs the task of assembling data file buffers from the Bus-Monitor data acquisition process and issuing these data files along with the appropriate control parameters to the Encoding Section in the real-time sequence defined to the LSI-11/23 by the Video Section.

One special function was implemented in a "spare" register address of the Video-Encoder that has no relationship to the encoding function. This function takes the form of a 16-bit read/write register latch with each bit wired to a readily accessible test point. This function was used for software development, testing, and measurement. This function was accomplished by the simple expedient of assigning each bit to a particular program task with insertion of bit-set and bit-clear instructions at the task entry and exit points respectively. Thus the register bit forms a task execution envelope at an "over-head" cost of about 3 microseconds for each task which is significantly lower than most other methods of inserting "test-probes" into real-time software. The connection of Logic Analyzers or other test equipment to these register test points allows very precise relative and absolute time measurements of the real time software execution.

The Video-Encoder video section (Figure 6) is functionally independent of the encoder section (Figure 7) in that all encoder functions are based on the timing and status outputs of the video section. One of the primary functions of the video section is to GEN-Lock the internal video sync to the external video source under computer control. The other primary function is to incorporate the TTL-level encoding signal from the encoder

section into the output video signal. One of the serious problems in incorporating the encoding component into the video signal is the actual coupling technique. Capacitive coupling results in serious bandwidth attenuation of the higher encoding frequencies. Direct coupling avoids the bandwidth problem but requires that the source video be at a known level to avoid serious video distortion that would cause problems for downstream equipment. Also, provision must be made for a video source that uses a capacitive-coupled output which will have DC component that varies with the average luminance level of the signal. Even a direct-coupled video source may have a DC offset due to relative grounding differential. The technique used to solve this problem was to use direct-coupling and to buffer the input into two channels. The first channel is a unity amplifier that is the input to DC Restoring Circuits that provides an offset correction to the second channel summing amplifier which thus normalizes the video source to local ground. The DC Restoring Circuity operates by sampling and filtering the level of successive sync tips. The restored video signal is routed to a video switch (discussed later) and the sync separator logic. The sync is separated from the restored external video by a comparator with a second input of sync-level (SLEV) which is the digital-to-analog conversion of a 6-bit programmable parameter. This allows a software algorithm the ability to autocalibrate for optimum GEN-Lock. A coarse horizontal stability test is also performed to determine if GEN-Lock should be attempted.

Control of the video section is primarily achieved by the software through the Video Control/Status Register. This register provides the ability to enable/disable the DC Restoring Circuit as well as the Video Switch and to initiate GEN-Lock busy cycles. The criteria for initiating GEN-Lock are that the Video Activity Detector (which is a rate-of-change detector) indicate activity and that the coarse horizontal stability test indicate "stable." When enabled, the GEN-Lock logic derives vertical and horizontal locking "resets" for the internal video sync generator (National Semiconductor MM5321) from the separated external sync. The horizontal locking level (HLEV) was programmable for horizontal locking from every horizontal sync to every 16th horizontal sync. However, it was found that locking every horizontal sync was required. When initiated,

the GEN-Lock busy cycle had three phases (normally taking 240 milliseconds) which are frame-lock, horizontal-lock, and lock-test. The lock-test function loads a programmable 8-bit fail-lock-level (FLEV) into a counter on each locking reset and uses the Exclusive-Or of external and internal sync to enable a count-down clock. If the counter underflows, GEN-Lock has failed and the locking resets are inhibited until another software initiated GEN-Lock cycle. If the lock-test phase passes, the video section remains in GEN-Lock mode with lock-test enabled until lock-fail or software lock-release. Another important feature is that the locking resets are "windowed" to limit the lock-tracking-rate to a range acceptable to the encoding process and to downstream video equipment. The video section also provides a maskable "Video-State-Change" program interrupt to the LSI-11/23. A Video-State-Change is defined as completion of a busy cycle or a change in the coarse horizontal stability test or the GEN-Lock mode (e.g., due to lock-fail) when the device is not-busy.

The final functional area of the video section is the video switch which has inputs of the external restored video and the internal video. The internal video is a black-field video signal (which may contain luminance-encoded data) generated from internal video sync and the serial data line from the encoder section by the Digital-to-Video Convertor (National Semiconductor LM 1886 and offset-scaling amplifier). The video switch "off" position applies the internal video signal to the output video buffer. When enabled by software and the condition of GEN-Locked, the video switch "on" position applies the external restored video to the output buffer. When the encoder section transmits serial data, it also asserts an over-ride "switch-off" for the period of horizontal rasters containing encoded data. This control logic for the video switch has several important consequences when the source video has scattered periods of unstable video. Two factors contribute to these consequences. The first factor is that, in the general case, downstream video equipment will lose GEN-Lock under the same conditions that the video section will lose GEN-Lock. The second factor is that downstream equipment (particularly VCR's) will regain GEN-Lock faster when subjected to a "step" discontinuity in vertical sync than when subjected to a period of general video instability followed by a return to stability. This is due to vertical sync GEN-Lock being primarily a "servo-loop-lock" function

which will ramp-in to a step discontinuity in sync but will phase-lag oscillate on general video instability. When the video section loses GEN-Lock, the downstream equipment sees no sudden change in video stability since the internal black field video will only slowly drift (due to clock error) from GEN-Lock with the last period of stable external video. Also when the external video stability returns and the software initiates a GEN-Lock cycle, downsteam equipment sees the GEN-Lock cycle as a step discontinuity in vertical sync. The first consequence of these conditions is that recoverable data encoding can continue even though unstable external video causes the loss of image data. The second consequence is that downstream equipment will tend to recover GEN-Lock faster when the video source stability returns.

The Encoder Section (Figure 7) of the Video Encoder performs the function of generating the formatted manchester-encoded serial data signal within the timing constraints of the internal video sync and the programmable parameters supplied by software. The programmable parameters fall into two groups (Encoder Parameter Register #1 and Register #2) that correspond to the two stages of encoder operation. Encoder Parameter Register #1 (EPR#1) selects a crystal reference frequency (24 or 14.31818 megahertz) and provides a 4-bit crystal divider that results in the encoding "phase-clock" (PCLK=2 X data clock). EPR#1 also provides an 8-bit value to a within-field horizontal sync downcounter whose underflow count determines the starting-horizontal-raster (FHGO=field horizontal go) for any encoding block. Encoder Parameter Register #2 (EPR#2) contains the encoder enable/reset bit. With encoder enabled, EPR#1 becomes read-only and the doubled format line is encoded in every video field unless the video section is executing the GEN-Lock function busy cycle. Encoder enable can be cleared by the fatal error of the selected encoding frequency being too low to allow encoding of a format line within a horizontal raster interval. EPR#2 also contains the two parameter fields (3 bits for data word-count/line and 8 bits for data line count) for controlling the insertion of encoded data raster lines between the two format raster lines. Once the encoder is set up and enabled, the actual data encoding process is controlled by the LSI-11/23 primarily through the Encoder Control/Status Register (ECSR). ECSR provides two

maskable interrupts to the LSI-11/23. The first interrupt is a selectable video-sync (start-of-field or start-of-frame) that is used to test for completion of any prior data encoding block and to initiate the next data encoding block if a data-output-file is ready. The second interrupt is data-encoding-done. Data is normally provided to the encoder via DMA loading of a FIFO buffer register but can also be provided by PCIO preloading of the FIFO buffer by the LSI-11/23. Data encoding is triggered by the LSI-11/23 (after initialization of the DMA address and word counter or after preloading the FIFO) via a "data-encode-go" command to ECSR. The encoder sequencer (which started format line encoding after "encoder-enable") tests for a condition of "data-armed" on the first horizontal sync within each video field in order to determine if data encoding raster lines are to be inserted between the format lines. Data-armed results from data-encode-go and the condition of FIFO stack output data ready (i.e., the FIFO contains enough data words to encode at least one full data line). As a result, after receiving the video sync interrupt, the LSI-11/23 has about 450 microseconds to initialize the data parameters and issue the data-encode-go to ensure that data encoding will occur within that field. If this constraint is not met, the data encoding will be delayed to occur in the next video field. The encoder sequencer performs the critical function of controlling the data flow sequence to the manchester word encoder as defined by the video timing and the encoding parameters. The sequencer controls various status flags and formatting information within the format raster lines and the command cell word that starts each encoded raster line. The sequencer also provides the video switch control to the video section that incorporates the encoding into the output video signal. It should be noted that if the FIFO stack has no data ready when the sequencer selects data, a "fill-word" will be substituted for data. Other than an error condition, this would occur only if the number of words provided for encoding were less than the number of words specified by the encoding parameters. This feature can be used to "hardware-normalize" a varying size data file to a maximum size video encoding envelope.

5. VIDEO DECODER INTERFACE ASSEMBLY

A block diagram of the Video Decoder is shown in Figure 8. This device is a BM/VE support interface that resides on the PDP-11/45 via the same Unibus Interface Assembly used by the 1553B Bus Simulator. The Video Decoder performs the function of recovering the digital data from the video output of the BM/VE proper. The PDP-11/45 sees the Video Decoder as a set of nine device registers and two interrupt request sources with acquired data normally transferred to main (PDP-11/45) memory by DMA. The two interrupts are selectable video sync (start-of-field or start-of-frame) and assertion of decoder-ready (i.e., data-collection-done). Data recovery by the Video Decoder operates on a per-video-field basis with a raw data structure of a file of word pairs consisting of a hardware-generated status word followed by the actual decoded data word. This status word specifies the actual horizontal raster line decoded and provides seven status flags. Three of these flags are for the hardware decoding errors of Manchester Format Error, Bit Count Error, and Parity Error. The other flags are Start-Of-Line, Empty Line, Command Cell, and Line Overflow Error. The Start of Line Flag is only set for the first word pair decoded on an individual raster line. The Empty Line Flag signifies that no encoding preamble syncs were detected on that horizontal raster line and that the following data word is a fill-word inserted by the Video Decoder. The Command Cell Flag is set if the decoded word meets the timing parameters defined for the command cell. The Line Overflow Error Flag is set if the video blanking over-rides a word being decoded, which will normally be due to video time instability. The raw data structure is processed by PDP-11/45 software acting on the decoding status word and the embedded (encoded) format information in order to complete the data recovery process.

The Video Decoder consists of Frontplane and Backplane subassemblies. The Frontplane provides the computer interfacing functions and provides programmable parameters (and controls) to the Backplane. The Backplane provides the functions of video processing/GEN-Lock, conversion of the manchester encoding to an NRZ format, and the reference clock logic that supports the manchester/NRZ conversion. As a debugging aid, the Backplane multiplexes the Frontplane parameters with Backplane parameter switches

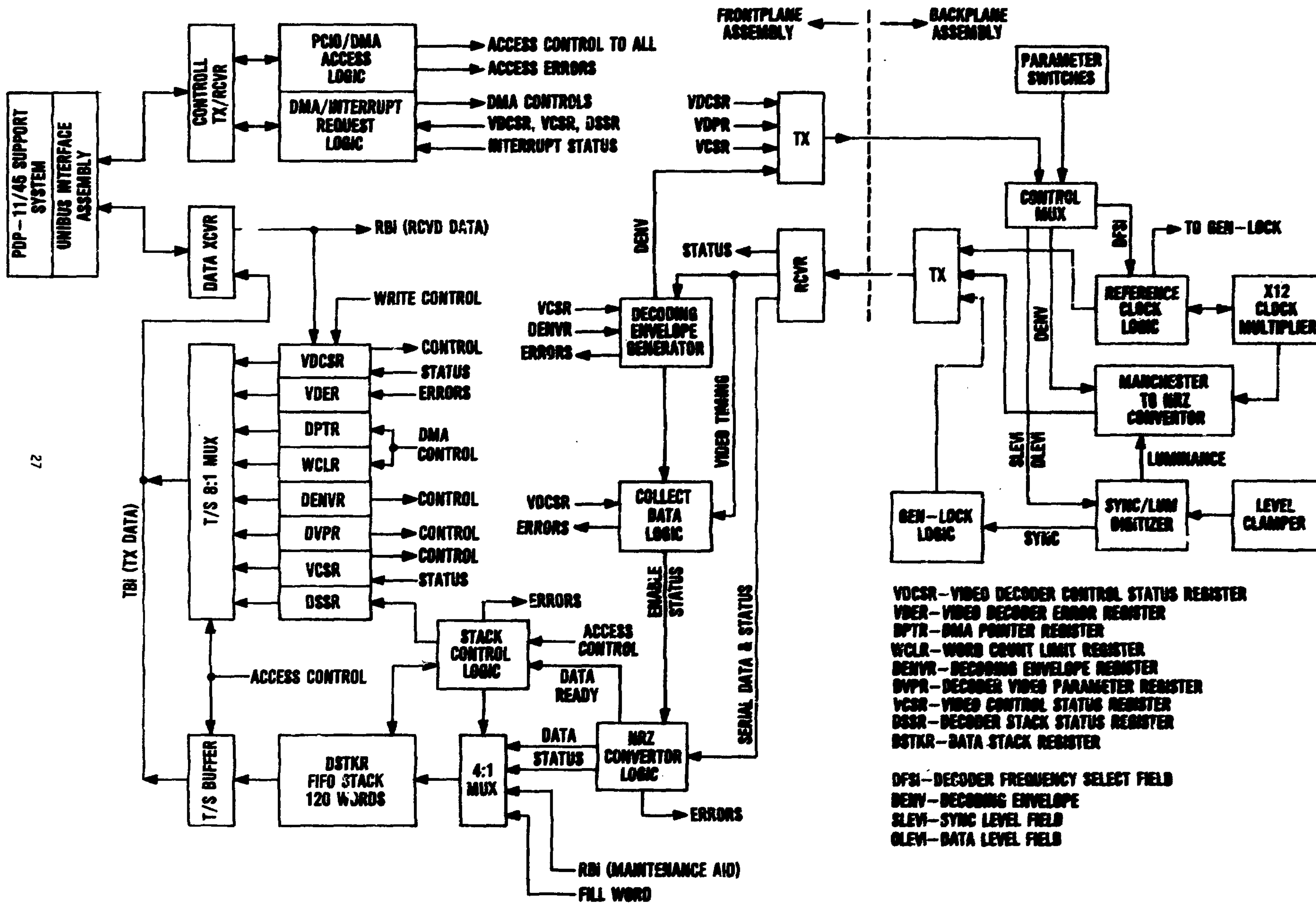


FIGURE 8. VIDEO-DECODER BLOCK DIAGRAM

which can provide an alternate control source that allows a form of independent operation for the Backplane. The Video Decoder has three basic stages of operation which are described in the following paragraphs.

The first stage of operation is video calibration. The input video is direct-coupled and terminated to 75 ohms. The input video is also applied to a buffer amplifier, a summing amplifier, and a video activity detector. The buffer amplifier output is applied to sample-and-hold logic that determines the voltage level of the video sync tips (with sampling derived at the horizontal frequency). This voltage level is scaled and applied to the second input of the summing amplifier. Thus the summing amplifier output is the input video clamped (sync-tip) to local ground. The video activity detector is level-independent logic that is slew-rate sensitive to typical video sync level transitions. The clamped video is converted to TTL-level luminance and sync signals via two voltage comparator circuits. The second input to each comparator is a digital-to-analog conversion of a 6-bit field (sync level or data level) provided by the Decoder Video Parameter Register. The Sync Level Field is nominally scaled to be variable from clamped sync tip level to the blanking level. The Data Level Field is nominally scaled to be variable from blanking level to 50% saturated white. These programmable values allow autocalibration software to obtain "best" performance. The separated composite sync signal is used to GEN-Lock an internal video sync generator to the external video signal. As long as video activity is detected, the GEN-Lock logic performs a coarse horizontal raster stability test, which if passed, applies horizontal locking resets to the internal video sync generator. Horizontal stability also enables a vertical locking reset derived from the fifth serration pulse of the video sync vertical interval. This logic can achieve GEN-Locked status in as little as 33 milliseconds and can hold lock even on the output of a fairly shaky video cassette recorder. The outputs of the internal sync generator are used for the overall video decoder timing and for the video sync interrupt regardless of locked/unlocked status. However unlocked status will inhibit or abort actual data collection with error flags.

The second stage of operation is video decoding without actual data collection. This stage is entered after PDP-11/45 initialization software provides three parameters and sets an enable bit. The first parameter selects one of 12 data decoding frequencies to match the BM/VE encoding rate. The other two parameters (8-bit fields in the Decoding Envelope Register) define the decoding envelope start line and decoding envelope line count. The decoding envelope generator uses these parameters (along with the internal sync generator outputs) to select the contiguous portion (or decoding envelope) of each video field that is to be processed by the manchester-to-NRZ convertor. The decoding envelope is additionally gated with horizontal blanking and provides a decode-enable to the manchester convertor with resets between each raster line during horizontal sync. Two error flags are associated with the decoding envelope. These errors are failure of the envelope to occur (if enabled) between vertical blanking intervals and failure by the manchester convertor of detecting any encoded word preamble sync waveforms within the decoding envelope. The manchester convertor decodes the digitized luminance into NRZ data using a biphase decoding clock at 12 times the encoded data rate. The decoding clock (up to 48 Megahertz) is derived from reference crystal oscillators (24.0 and 14.31818 Megahertz) by divider/Phase-Lock-Loop logic (using Signetics NE564 PLL oscillators). The phase detectors of the NE564s are also used to generate decoding frequency stability status signals. As a result the manchester convertor can operate at 12 data rates ranging from 1.19 to 4.00 Megahertz. The manchester convertor design was based on quantizing the luminance signal into an 8-bit data shift register at the decoding clock rate (12 x data rate). The first two stages of the data shift register acted as a transition finder for a 16-bit transition shift register with supporting logic testing transition validity. The validity testing (after the word start transition) was always relative to the prior transition with an allowable "window-variation" of three decoding clocks. The transition logic would also insert one missing transition (e.g., the missing phase transition for manchester data "1" followed by data "Ø") but would declare a manchester format error on two sequential missing transitions. The transition shift register also operates as an activity-detector/word-time-out function. The transition shift register logic also provides

"data-windows" and "phase windows" that are gated with the outputs of a directional transition finder (operating on the last two stages of the data shift register) to separate the manchester signal into data and phase transitions for "black-to-white" and "white-to-black" (logic 1 or "up" and logic Ø or "down" respectively). The phase transitions are then ignored within a word-decoding envelope except for testing that each windowed data/phase transition is followed by an opposite polarity windowed phase/data transition or else the second type of manchester format error is declared. When enabled, the manchester convertor has two states. In the first state, the convertor searches for the encoded preamble sync wave form sequence of "data-up/phase-down/phase-up/data-down." When the sync preamble is detected, the actual manchester-to-NRZ conversion state is entered with the assertion of word-envelope. The conversion state uses the windowed data transitions to generate the NRZ serial data and data clock. The word-envelope is terminated by manchester format error (which includes transition time-out) or by receipt of 17 decoded bits (16 data bits + parity bit). The convertor then returns to the preamble-search-state. The convertor outputs to the frontplane are:

- Transition activity status
- Preamble-sync detected status (pulse)
- Word (decoding) envelope
- Serial NRZ data
- Serial data clock (gated by word envelope)
- Bit-Count-OK flag
- Manchester format error flag

The third stage of Video Decoder operation is the actual data collection which includes serial-to-parallel conversion and parity checking. Data collection occurs only on a "go" command from the PDP-11/45 which is normally "triggered" by the video sync interrupt. Data is collected from the first complete decoding envelope that occurs after the go command. The collected data is normally transferred to the PDP-11/45 via DMA, but small raw data files containing 60 word-pairs or less may be accumulated in the FIFO Data Stack Register to be transferred by direct PCIO access. Prior to issuing a DMA-Enable/Go, the PDP-11/45 must

initilize the DMA Address Pointer Register and the Word Count Limit Register. The Word Count Register is a "Limit"-type because the size of the raw data file cannot be assumed to be fixed but an error flag will be set on count-overflow that inhibits further DMA writes to PDP-11/45 memory. Once started, the collect-data-logic and NRZ convertor generates at least one word pair (hardware decoding status word and decoded data word) for every horizontal raster within the decoding envelope. If a particular raster has no detectable preamble syncs at the end of that raster, a word pair is entered into the Data Stack Register with the Empty Line Status Flag set and a fill word in the decoded data position. Otherwise, each preamble sync within a raster generates a word pair entry with appropriate flags set in the hardware status word and the decoded data (or the right-justified portion of the data received prior to a hardware decoding error). The data collection cycle is terminated after completion of the decoding envelope and any pending DMA cycles required to empty the Data Stack Register.

The Video Decoder recognizes 15 error conditions that have individual flags in the Video Decoder Error Register. Four of these error flags act as a summary of all decoded rasters for the hardware decoding status word errors of Line Overflow Error, Manchester Format Error, Bit-Count Error, and Parity Error. Another summary-type error is provided to flag the condition of a horizontal raster with decoded data that had no detectable command cell. The remaining 10 errors directly affect data collection and these errors fall into three classes. The first class is a sequence error which aborts the collection cycle if the go-command was issued during video or decoding clock time instability. The second class of errors terminates DMA but allows data collection to continue within the storage limits of the Data Stack Register. The second class of errors includes DMA access time-out, word count overflow, and any attempt to DMA-write to the PDP-11/45 operating system memory space. The third class of errors terminates data collection but allows completion of any pending DMA cycles. The third class of errors can occur only within the decoding envelope and include video stability lost, decoding frequency stability lost, no detectable preamble sync waveforms (by end of envelope), data stack overflow, and any decoding envelope timing error.

SECTION III

VIDEO ENCODING FORMAT

The video encoding format implemented in the BM/VE is the direct result of our experience in the flight test programs mentioned in Section I.1. The problems of data recovery from video recorded on an airborne VCR can be simplified into three categories. The first category is vertical instability which primarily results from aircraft acceleration affecting the VCR tape tensioning mechanism. The second category is horizontal instability which primarily results from aircraft vibration affecting VCR tape skew. The third category is tape imperfections which can cause momentary video dropout which can seriously effect video encoding without significantly degrading the video image. Even when these problems do degrade the video image, this degradation can usually be improved by using a Video Time Base Corrector (VTBC) with the playback VCR. While the VTBC will cause some loss of data due to quantification errors, there is a net gain in data recovery that results from the improved video stability. The combination of VCR and VTBC also has three troublesome tricks when correcting for video instability that must be provided for in the data recovery process. These tricks occur because the VTBC either averages out instability or delays the instability to the bottom of each video field which is normally off-screen on a display monitor. The first trick is that small vertical instability may be corrected by "field-reversing" the video sync relative to the image data. Also during normal operation the VTBC will delay all luminance data including the encoding envelope from the nominal position within the video field by several vertical raster positions. The second trick is that the VTBC will compensate for instability by varying the size of this delay between successive video fields. Typically, the position of the encoding envelope can jitter as much as two vertical positions. The third trick occurs for relatively large horizontal errors like those that occur on momentary dropout. The VTBC might correct the instability by repeating the prior horizontal raster or by decreasing the internal delay by one raster line within that field. Thus, the data recovery process can see the encoding envelope vary in position and decrease in

size with the missing raster line even being from the middle of the envelope. It can even see a normal size envelope with one line missing and one line repeated. These problems are in addition to the more normal recovery problems of noise, signal distortion, and reference clock stability/accuracy which tend to be evenly distributed throughout the encoding envelope. Therefore the key formatting characteristics should have the features of:

a. A unique video field identifier with redundancy to increase the probability of recovery

b. A unique line identifier in each encoded raster within the encoding envelope

c. Independent word encoding so that recovery errors do not propagate forward

d. Status information to improve error recovery

e. The use of manchester encoding to reduce sensitivity to time instability

While these characteristics increase the encoding overhead percentage, our experience shows it to be a worthwhile investment. Two additional features are required in the data recovery process. The first feature is that encoded word detection should not require error free word recovery but should flag recovery errors. The second feature is that the recovery process must be capable of handling a horizontal raster with no detectable encoding even though encoding may have been expected on that raster. This feature allows the data recovery process (Video Decoder) to "over-decode" or in other words to use a decoding envelope several rasters larger on each end than the expected encoding envelope. Thus, the encoding envelope will remain within the decoding envelope even if the encoding envelope has a vertical positional jitter.

The basic unit of encoding is the word cell. The word cell format (with examples) is shown in Figure 9. The word cell contains 20-bit cells of encoding with a minimum interword gap of 2-bit cells containing no activity. The word cell consists of 3-bit cells of preamble sync

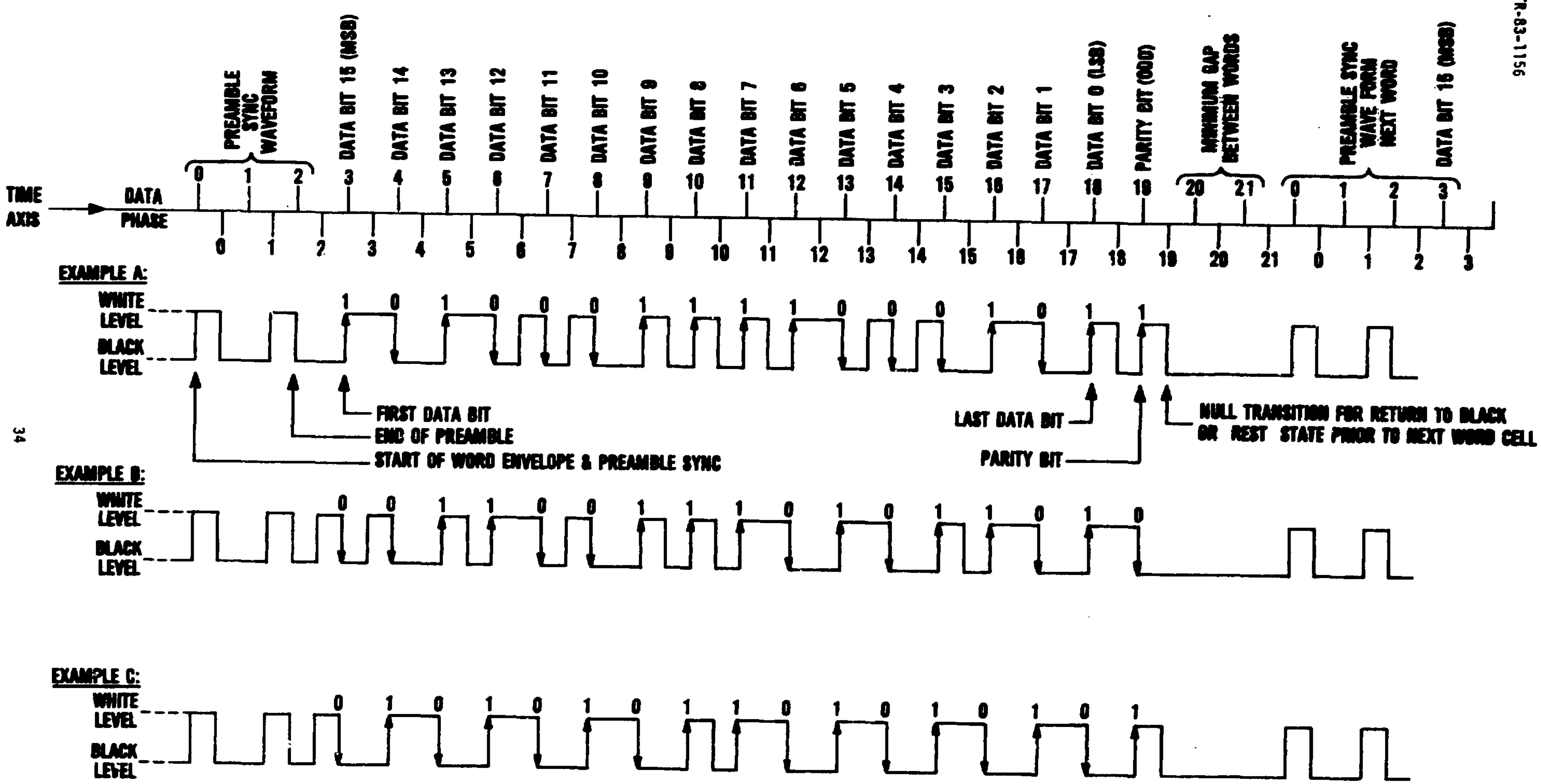


FIGURE 9. WORD ENCODING FORMAT WITH THREE EXAMPLES

followed by 16-bit cells of data (most significant bit first) followed by a parity bit cell. The parity bit and the 16 data bits provide for an odd parity check on data recovery. The preamble sync is an invalid manchester pattern which is used for word detection by the Video Decoder. The actual video encoding is unipolar luminance transitions between the video black level and the (nominal) saturated white level. In 1-volt RS-170 composite video, this black-white transition is nominally 0.5 volts. The black-to-white (or "UP") and white-to-black (or "DOWN") transitions are respectively defined as logic one and logic zero. As with normal manchester, transitions at the "phase" point (halfway between data points) are for setup of a data transition of the same polarity as the prior transition. The 3-bit cell preamble sync consists of the sequence data-up (the first transition after an encoding gap is defined as a data transition) followed by a phase-down followed by a "missing data-up" followed by an actual phase-up followed by a data-down. An alternate way to describe the preamble sync is that it is a manchester 110 with the second data transition delayed to the phase point. This invalid manchester sequence following an encoding interword gap (of two bit cells or more) with the signal at the "rest" or black level is readily identifiable as a start-of-word to the Video Decoder. Since the rest condition for encoding is defined as black level, when the encoding parity bit is a logic one (data-up), it is necessary to follow the parity bit one with a "null" transition of phase-down to return the video level to the rest state. As shown in Figure 9 Example A, this null transition does not impinge on the 2-bit cell minimum interword gap.

The encoding envelope has two levels of formatting. The first level is that all encoded word cells on any individual raster are preceded by an encoded command cell. The format of the command cell is given in Figure 10. The command cell specifies how many word cells follow the command cell on that raster line and contains a unique identifier for each raster that also functions as an encoded raster countdown value that specifies the remaining number of encoded rasters within the encoding envelope. The command cell contains two flags to indicate video status and one flag to indicate an encoding abort at the end of the current raster due to any data/timing errors within the BM/VE. The command cell

ENCODED COMMAND CELL CONTENTS			
BIT	NAME	FUNCTION	NOTES
15	AFLG	ABORT FLAG 1 = DATA ABORT/ERROR	1
14	DFLG	DATA FLAG 1 = DATA RASTER LINE	
13	FFLG	FORMAT FLAG 1 = FORMAT RASTER LINE	
12	VOKTE	VIDEO OK TO ENCODE = 1	2
11	GLOK	GEN-LOCK OK = 1 Ø = UNLOCKED	3
10	EWC2	ENCODED WORD COUNT	4
9	EWC1		
8	EWCØ		
7	RLC 7	REMAINING LINE COUNT	5
6	RLC 6		
5	RLC 5		
4	RLC 4		
3	RLC 3		
2	RLC 2		
1	RLC 1		
0	RLC Ø		

NOTE 1: This indicates a BM/VE data/timing error affecting data recovery.

NOTE 2: VOKTE = Ø indicates that video stability is suspected/bad and will abort any encoding in progress.

NOTE 3: GLOK = Ø indicates that external video is bad and that the BM/VE is substituting internal stable black-field video.

NOTE 4: EWCi specifies the number of encoded word cells following the command cell on any individual raster. EWCi values range from Ø words to 7 words.

NOTE 5: RLCi is a countdown of the remaining encoded rasters within the encoding envelope. The terminating line of each encoding envelope will be a format line with RLCi = FFH.

Figure 10. Command Cell Format

also has two flags that identify that encoding raster as a data raster or as a format raster. The format raster is the second level of format structure within an encoding envelope. The format raster is repeated twice within each encoding envelope as the first and last rasters within the envelope. The only difference between the two format rasters is in the Remaining Line Count in the command cell on each raster. If no actual data is encoded in a particular video field, the two format rasters are contiguous. Figure 11 gives three examples of video fields with encoding envelopes. A format raster contains three encoding cells consisting of a command cell followed by two encoding cells containing Format Line Register #1 (FLR#1) and Format Line Register #2 (FLR#2). The format registers contain a 20-bit video field counter that provides the video encoding time reference. The field counter increments with each (video) vertical drive unless the BM/VE is attempting to acquire GEN-Lock to the external video. The field counter is also synchronized to identify the even/odd video fields. FLR#1 provides the capability to define a "data file" of between one and 16 continguous fields, each field containing a data subfile within the encoding envelope. The data file structure information consists of a 4-bit subfile initial value and a 4-bit subfile counter in addition to two flags for start-of-file and end-of-file. The contents of the format line registers are given in Figure 12.

The encoding format provides support for several different types of data files. Large data files may be encoded in multiple subfiles or alternatively as a large encoding envelope that "steals" a video field at some interval such as once a second. Data files can be encoded after acquisition at less than 60 hertz since encoded data need not be incorporated in every video field. Also unneeded image data can be sacrificed to allow encoding envelopes that are larger than the horizontal portion of the vertical blanking interval. While this format has an overhead that could typically run about 25% of data, the formatting information readily supports data recovery processes that give a high probability that loss of individual data words will not cause loss of the entire data file.

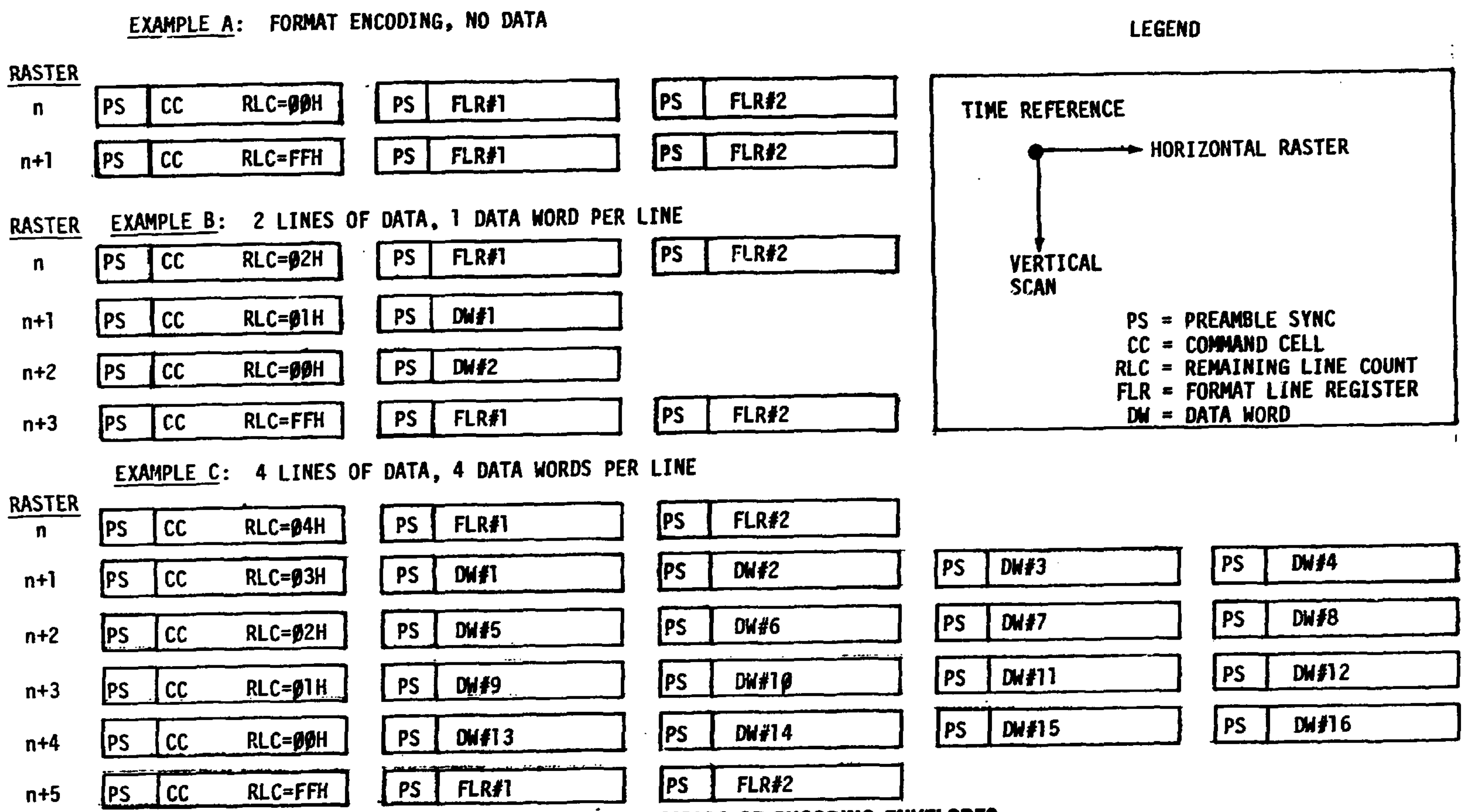


FIGURE 11. EXAMPLES OF ENCODING ENVELOPES

FLR1: FORMAT LINE REGISTER #1

BIT	NAME	FUNCTION	NOTES
15	SODF	START OF DATA FILE = 1	1
14	EODF	END OF DATA FILE = 1	
13	--	SPARE = 0	
12	--	SPARE = 0	
11	SFC3	SUBFILE COUNTER	2
10	SFC2		
9	SFC1		
8	SFC0		
7	SIV3	SUBFILE (COUNTER) INITIAL VALUE	3
6	SIV2		
5	SIV1		
4	SIV0		
3	FC19	MSB VIDEO FIELD COUNTER HIGH 4 BITS	
2	FC18		
1	FC17		
0	FC16		

FLR2: FORMAT LINE REGISTER #2

BIT	NAME	FUNCTION	NOTES
15	FC15	VIDEO FIELD COUNTER LOW 16 BITS	4
14	FC14		
13	FC13		
12	FC12		
11	FC11		
10	FC10		
9	FC9		
8	FC8		
7	FC7		
6	FC6		
5	FC5		
4	FC4		
3	FC3		
2	FC2		
1	FC1		
0	FC0	LSB	

NOTE 1: For data files containing a single encoding envelope, SODF and EODF will both be set.

NOTE 2: SFCi provides subfile identification and functions as a countdown indicator of the remaining subfiles within the data file. SFCi is loaded with the value of SIVi at SODF=1.

NOTE 3: SIVi defines the number of encoding envelope subfiles (fields) in a data file.
For SIVi=0H: 1 data file = 1 encoding envelope.
For SIVi=FH: 1 data file = 16 encoding envelopes.

NOTE 4: FCi is video frame synchronized so that FCi is odd for the first field within a video frame.

FIGURE 12. FORMAT LINE REGISTERS

SECTION IV

DEVELOPMENT SOFTWARE

The software for the Bus Monitor/Video Encoder is broken into four major categories: Support Software (Section IV.1), BM/VE System Software (Section IV.2), Bus-Monitor Software (Section IV.3), Video-Encoder Software, (Section IV.4), and Software Summary (Section IV.5).

The support software consists of all software which is used for the generation of test data and the verification of the performance of the BM/VE system or subsystems. Primarily, there are two major programs required to support the development of the BM/VE system, the 1553B bus simulator and the video decoder.

The BM/VE system software consists of the necessary programs to support operations of the BM/VE system. These programs include a terminal communication link program and special loader used to downline load the BM/VE system through an RS-232 link, an EPROM Programmer program, an 8086 cross-assembler and linker, and 8086 loader program, and necessary device drivers to support the I/O to the various devices supporting the BM/VE.

The Bus-Monitor software is the operational 8086 software in EPROM. The 8086 cross-assembler and linker programs were obtained from a Digital Equipment Corporation Users Society (DECUS) Structured Languages SIG tape and were written in Pascal. The loader program is written in RATFOR and runs in the LSI-11 under an RSX-11S operating system. The EPROM programmer program also is written in RATFOR and runs in the RSX-11S system in the LSI-11.

The Video-Encoder software is the operational LSI-11 software which controls execution of the bus-monitor software and performs the data packet encoding control function. This software is completely PDP-11 (LSI-11) MACRO-11 assembly code. The code was developed on the PDP-11/45 RSX-11M system and downloaded into the LSI-11 via a custom loader program, also written in MACRO-11 and resident in the EPROM in the BM/VE.

The software developed for the BM/VE was developed on a PDP-11/45 using the RSX-11M operating system. All support software was written in RATFOR (a structured Fortran pre-processor). The software development process for the BM/VE system was a multi-phase operation. Initially, test code was written in RATFOR and tested on the BM/VE under the RSX-11S operating system. This code was used for the hardware debugging support and algorithm development. As the hardware and code were debugged, the code was rewritten in MACRO-11 and added to the operational BM/VE code. The final operational BM/VE code is the culmination of many iterations of the above process.

The only code which was not first written in RATFOR is the 8086 code which was written in assembly code from the start. The debugging process for this was slightly different. The 8086 code was initially loaded into RAM for debugging purposes. When the code was operational in RAM and fully debugged, it was re-linked to be loaded into EPROM. This became part of the final version of the bus-monitor code. This process eliminated unnecessary erasing of the EPROMs to burn in new code, since the only code burned into the EPROMs was fully debugged.

The 8086 assembly language code is cross-assembled with a cross-assembler written in Pascal. This cross-assembler was acquired from a DECUS Structured Languages Special Interest Group (SIG) tape. This tape had both the cross-assembler and the linker on it, both written in Pascal. This sytem provides for the separate assembly and subsequent linkage of source modules. The assembler fully supports the 8086 instruction set.

Any necessary device drivers for the RSX-11M operating system were written in MACRO-11 by Mr. William Mikulski. Operational debugging of the BM/VE code was greatly aided by use of a Hewlett-Packard Logic State Analyzer (HP-1615A). The analyzer was used for debugging of both hardware and software, and was used for passive software timing.

1. SUPPORT SOFTWARE

In order to test the BM/VE functions, certain hardware and software had to be built and generated. One piece of hardware required was a 1553B bus simulator. This was necessary to provide 1553B data for

debugging the Bus-Monitor function of the BM/VE. After the 1553B bus simulator hardware was built, software was required to generate various data bus profiles. For this, Mr. Mikulski generated several programs to support the 1553B bus simulator. One of these programs generates a data file containing the necessary control and data words to output to the 1553B bus simulator itself. The 1553B bus simulator requires a specific format for the control and data words sent to it. This program generates the data files which conform to this required format. The program allows for varying any of the controllable parameters, such as channel selection, number of data words, terminal ID, forced parity errors, etc.

The other necessary program was one which reads the data file created by the above program and outputs it to the 1553B bus simulator itself. This program allows for up to three different files to be accessed and output in sequence to the generator. In order to provide for minimum system overhead, the 1553B bus simulator allows for continuous DMA operation. This means that the program can start the simulator in DMA loop mode and then wait for an error or a signal to terminate the transfer. This provides for a very high data rate transfer to be sent over the 1553B bus. Alternatively, the program can output the data files on a scheduled basis, allowing for total program control of the interval between file transmissions. This mode generates a profile which has high data burst rates at a lower interval than can be obtained with DMA loop mode.

In order for the program to operate in the RSX-11M environment, a device driver had to be written to interface between the user program and the hardware registers in the 1553B bus simulator. This device driver has to support both program mode I/O and DMA operations. Since the hardware can support continuous loop mode DMA, the driver has to be nonstandard, in that I/O can be happening after the driver returns a successful completion code to the user. Also, if an error occurs after initiation of the loop mode DMA, an interrupt occurs to the driver with no I/O packet queued, which will crash the operating system. These details were resolved and the final driver successfully supported the required operations of both programmed I/O and continuous loop mode DMA.

The other end of the system required a means of validating the encoded data. This required the design of a Video Decoder. Similarly, the video decoder required software to acquire the data from the decoder and to analyze that data to determine the validity of the decoding. Due to time constraints, only limited validation was done. The validation program consisted of code to snapshot blocks of decoded data and count data errors. The program initialized the decoder with an automatic sync level setup procedure. Then the decoder parameters were set based on user defined values. These parameters determined the starting line and number of lines to decode. Using these parameters and an assumed value of three data words per line, the total number of data words expected were computed. This was used in computing the percent data error rate. Tests were performed in four modes: 1) raw video from the encoder, 2) raw video from the encoder processed through a time base corrector (TBC), 3) recorded video, and 4) recorded video processed through the TBC. The results are summarized in Section V. It was experimentally determined that the error rate was significantly reduced if the decoding window started one or two lines before the actual encoding started.

2. BM/VE SYSTEM SOFTWARE

There are many support programs involved in the BM/VE system. The first program required to support the BM/VE directly is the terminal communication program resident on the host system, the PDP-11/45. This communication program is a combination of RATFOR code and MACRO-11 code. The program is more than just a terminal emulator or terminal link program. It has the capability to redirect terminal input and/or output to a disk file. Also, input disk files can be down loaded to the LSI-11 via the micro ODT in the LSI-11. Using this program, a pre-loader is loaded into the LSI-11 which is subsequently used to download a full RSX-11S system into the LSI-11. This second loader was burned into EPROM so the load process consists of setting up the memory management registers to map to the EPROM and then executing the loader code. The loader code was written in MACRO-11 and is position-dependent, using no RAM for scratch, only registers, and not even a stack. This allows the code to execute out of EPROM to fully load the LSI-11 RAM without requiring any

stack space. The loader program is status driven, and provides for block loading with a validity checksum returned to the PDP-11/45 upon completion.

The terminal link program can load the standard LSI-11 diagnostic programs from disk files for execution in the LSI-11. These include CPU exerciser, Memory Management exerciser, Memory exerciser, and Floating Point exerciser. Another special file type which can be downloaded is an 8086 load module. This is a text file which is an ASCII Hexadecimal dump (essentially) of the region of memory to load. This file is generated by the 8086 linker. This code can be sent to the LSI-11 by the terminal link program, but must be read by the 8086 loader program running in the LSI-11. As indicated above, the link program can also load an RSX-11S system image file to the LSI-11 either with the LSI-11 loader program running or with the micro ODT in the LSI-11. The link operates split rate, 9600 baud transmission to the LSI-11, 1200 baud transmission to the PDP. This allows fast loading of the LSI-11 with the EPROM resident loader; a full 64KWord RSX-11S system can be loaded in about 2.5 minutes.

The terminal link program is essentially an AST driven processor. Asynchronous System Traps (AST's) are handled by the program as they occur. AST's occur for input from the user keyboard or from the LSI-11, and for output to the user display and the LSI-11. Each of these AST's sets a unique local event flag, which is detected by the main program for subsequent conditional processing.

The normal mode of communication for the terminal link program is interactive communication. In this mode, the user's terminal appears to be directly connected to the LSI-11. Certain special characters are detected by the AST handler in the program and special event flags are set to indicate which special character has been typed. Upon detection of one of these event flags, the main program calls a special subroutine to handle the request. These requests are typically for loading of files, or for exiting the link program. In order to handle the I/O rate, the link program must run at software priority 250 under RSX. This is the highest priority available with the RSX-11M operating system. Due to the large overhead of terminal I/O, any lower priority results in loss of characters from the LSI-11.

In order to load the 8086 load modules, there is another program which runs in the LSI-11 in the RSX-11S sytem, the LOAD86 program. This program reads a load module from the input (loaded from a file by the link program), decodes the hexadecimal dump format, and stores the decoded values in 8086 memory. This dual-ported memory is mapped as a resident common in the RSX-11S system, and mapped to in the LOAD86 program. Additionally, if the LOAD86 program detects that the address being loaded is actually in the EPROM, it attempts to program that address with the desired contents. If the programmer power supply is not on, or an error occurs during programming, the BM/VE system must be shut down and restarted from power-off. This prevents inadvertently over-writing good code in the EPROM.

3. BUS-MONITOR SOFTWARE

The Bus-Monitor Software is the 8086 code which is used to perform the data collection from the 1553 data bus. The code allows for selective device monitoring, and automatically discards messages which contain no data, unless there is an error. The bus-monitor code is resident in the EPROM for immediate execution by the 8086 upon being enabled. All data for both control of the monitoring and for output from the decoder is located in the shared RAM area.

Information which is required by the monitor software at startup is the current date and time of day (used to initialize IRIG clock), and a list of devices to monitor. This information is located at fixed addresses in the shared RAM. The Video-Encoder Software initializes these locations when it starts the 8086 the first time. After that, the locations are assumed to be correct.

The bus monitor code consists of an initialization section, and several interrupt code blocks. The primary operation of the monitor is via a hard coded monitor loop which continuously monitors the status of the 1553 decoder hardware. When there is data available in the stack, the program reads the data in a loop until the stack is empty. During this read cycle, the program is monitoring the buffer contents to determine if the last buffer read had any data in it, or if there are any

errors. If there is no data and no errors, the buffer point is reset to the start of the last data packet, thus ignoring empty, error-free messages. When the data buffer overflows, the input buffer is switched, and the LSI-11 is informed of the overflow via an interrupt request. The bus-monitor program then continues its monitoring operation, first by retrieving IRIG time and placing it at the beginning of the new buffer. By doing this, each buffer of data is time tagged with the current time, accurate to the nearest 1/100 second.

The LSI-11 can also interrupt the 8086 for special processing. There are four interrupts available from the LSI-11. These can be used for tracing execution of the 8086 code, or for altering or reinitializing the monitoring functions. The devices which are being monitored are loaded into the decoder hardware registers only upon initialization of the 8086 monitor code. Therefore, in order to deselect or select a new device, the device code bit must be cleared or set in the common RAM area, then the monitor code must be reinitialized. This is done with one of the available interrupts. Another interrupt is used to take a snapshot dump of the 8086 registers. This is used for debugging the 8086 code. On interrupt, the 8086 saves all of its registers in locations in shared RAM, available for inspection by the LSI-11. It also saves the IRIG time in shared RAM. The other two interrupts from the LSI-11 are not used.

There are two interrupt lines from the 8086 to the LSI-11. Interrupt request A is used for any error detected by the 8086. Interrupt request B is used for indication of buffer overflow and subsequent switching of data buffers by the monitor program in the 8086. Error interrupts are identified by the 8086 before asserting request A. An error type flag is loaded into a location in shared RAM, which is examined by the LSI-11 upon receipt of interrupt A. This way, a common 8086 error handler can be used for all errors. Specifically, an occurrence of an error, the 8086 places the error code into the A register then calls the error save code. This code saves all registers, IRIG time, and places the error code from the A register into the flag location in shared RAM, then sets interrupt request A. Depending on the type of error the 8086 will either halt or return to its normal processing via a restart operation.

In addition to the two software interrupt requests, the hardware status of the 8086 is tied to two more interrupts, the stopped interrupt, and the error interrupt. If the 8086 stops for any reason, or encounters an error, the corresponding request is set. With the use of these four interrupts, complete status of the 8086 monitor software is available to the LSI-11.

4. VIDEO-ENCODER SOFTWARE

The Video-Encoder Software is the heart of the BM/VE system software. All operations are controlled by this software. The video-encoder software is an interrupt driven operating system written in MACRO-11. It is task built as a system under the RSX-11M operating system on the PDP-11/45. It is downloaded into the LSI-11 from the PDP-11 via the terminal link program and the LSI-11 resident checksum loader in EPROM.

The video-encoder program is a custom real-time operating system. It has the capability of examining and modifying any memory location in the LSI-11 or the shared RAM. When the program is loaded and initiated, there are two basic modes of operation: 1) Video-Encoder control mode and 2) 8086 control mode. In both modes, all commands are single-letter commands entered from the RS-232 terminal link. The two modes are toggled back and forth with the ESCAPE key. In each mode, a '?' will display a HELP page with brief explanations of each command available. Any number which is entered is placed into the increment register. This register is actually a memory location, not a CPU register, and is used for all of the video-encoder parameter modifications. This register is also used for examining and modifying memory locations. All output to the terminal is via a large FIFO ring-buffer. This buffer can be flushed with the 'Q' (Queue) command. Using a FIFO queue assures that no messages will be lost due to timing problems. There may, however, be a significant delay between the occurrence of an event and the eventual display of a corresponding message if the queue is not nearly empty.

The command parser was written in such a way that commands could be added or deleted easily. This allowed for addition of debugging code with special temporary command codes if necessary. The command parser

was essentially a lookup table generated offset to a process dispatch table. One of two process dispatch tables was used, depending on the mode (encoder control or 8086 control).

5. SOFTWARE SUMMARY

The BM/VE software is a combination of higher-order language programs for support (RATFOR) and assembly language programs and systems for the real-time operations. The design methodology of the assembly language programs was one of high-order design and debug followed by assembly language coding. This maintained the overall structured design of the software while allowing maximum efficiency and control from an assembly language system.

The design of the software was considered in the design of the hardware and vice versa. The software designer/programmer was an integral part of the hardware design process. No hardware was designed without consulting with the software designer throughout the design process. This assured a smooth interface between the software and the hardware. All too often, hardware is designed without any consideration of software impact, with the result that one ends up with a highly capable piece of hardware that is a software designer's nightmare.

SECTION V

EXPERIMENTAL RESULTS

The experimental results may be broken into four categories which are BM/VE software parameters, data recovery (Video Decoder) software parameters, Bus Monitor hardware performance, and Video Encoder/Decoder hardware performance. Each of these categories is discussed in the following paragraphs. The software parameters are purposefully provided as "ballpark" figures since the project objective was to obtain general requirements rather than specific processor performance. Ballpark figures are also used because no widely accepted benchmark testing is directly applicable to the BM/VE real-time software operations. Numeric results are given for the hardware performance.

The BM/VE software breaks into one section for the Bus-Monitor microprocessor (8086 with 6 Megahertz clock and 700 nanosecond access-cycle memory) and one section for the BM/VE main microprocessor (LSI-11/23 with 400 nanosecond access-cycle memory) which handles overall synchronization and the video encoding task. It was found that the communications interlocking between the LSI-11/23 and the 8086 only provided less than 2% of the task loading which validates the approach used of cross-linked program interrupts and a dual-port memory. The program for the LSI-11/23 required about 4 KWords of memory. The 8086 program required about 1.5 KWords of program memory. The LSI-11/23 and 8086 also shared a data buffer of 1 KWord and a control parameter buffer of 40 words. Processor loading of the LSI-11/23 was about 25%. However the program execution pattern of the LSI-11/23 was high burst rate followed by idle periods since the video encoder software and 8066 communications involved 120-180 program interrupts per second. As a result, the system synchronization and video encoding software requires processor capabilities on the order of 250-300 Thousand Operations Per Second (KOPS) with an average interrupt latency of about 30 microseconds. While the real-time constraints of the LSI-11/23 software are not compatible with Higher Order Language (HOL) programming, it would be feasible to use a software operating system that had provision for direct interrupt and

input/output connection to a user assembly-level program. The processor loading of the 8086 Bus-Monitor software execution reached 100% for the higher data acquisition rates. The Bus-Monitor software requires processor capabilities on the order of 350 KOPS with an average interrupt latency of about 20 microseconds. In pragmatic terms, the real-time constraints for the Bus-Monitor software virtually mandate the use of carefully constructed assembly level programming.

The data recovery software requirments can be met with about 4 KWords of program memory and 4 KWords of data buffer memory. The data recovery process is compatible with use of a software operating system and HOL programming. The data recovery software requires processor capabilites on the order of 200 KOPS with an average interrupt latency of about 130 microseconds. Two approaches can be used for data recovery. The first approach acquires the raw data in real-time with off-line storage (e.g., magnetic tape) for later non-real-time data normalization and reduction. The second approach combines raw data recovery and data normalization with off-line storage in real-time which requires higher processor throughput.

The Bus-Monitor hardware was tested for two modes of operation. In the first mode all 1553B bus traffic was acquired and stored in memory. In this mode, the Bus-Monitor reach 100% real-time loading at a data acquisition rate of 38,000 words per second. In the second mode, the acquired data is time-tagged and sorted to delete the 1553B polling message overhead (which is typically 30% of 1553B traffic). In the tag/sort mode, the Bus-Monitor reached 100% real-time loading at a data acquisition rate of 31,700 words per second. While data acquisition errors (forced by the 1553B Bus Simulator) would cause loss of individual messages, these errors had no significant impact on maximum data acquisition rates. In our opinion, the theoretical maximum 1553B rate of 48,000 words per second could readily be achieved by upgrading the existing Bus Monitor with 350 nanosecond memory and an 8086 with 10 megahertz clock. In practical terms, the current Bus Monitor has over five times the instrumentation bandwidth available for use in a video signal without a major sacrifice of image data. For example, with maximum data acquisition and a video encoding data rate of 4 megahertz, about one half of the active image data would be replaced with video encoding.

In terms of experimental results, the video encoding and data recovery processes (or Video Encoder and Video Decoder) are quantitatively inseparable. Two additional factors should be kept in mind when considering these results. The first factor is that the video circuits were all implemented on wire-wrap socket assemblies with all the associated problems of parasitic capacitance and noise coupling. As a result, the video signal had a noise-to-signal ratio of about 15% and the Video Encoder/Decoder video bandwidths were about 8 megahertz. In our opinion, the first factor also contributed to the failure to hold GEN-Lock with a High Average Picture Level (HI-APL) video signal due to the worst case slew rates around the video sync regions. Video GEN-Lock was successfully held for all other categories of black/white or color video (RS-170 or NTSC) either direct or capacitively coupled from signal sources of a video test generator, a Hughes CCD camera, and a VCR/VTBC. The second factor is that all crystal clock reference frequencies had a relative accuracy of 0.017% with a short term stability of 0.004%. The first result of the clock accuracy is that the GEN-Lock function had about 1.5% time base jitter in the horizontal raster. The second result of the clock accuracy is that data recovery decreased due to quantization errors as the decoding clock (12 X data rate) approached the "logic-gate-delay" time of the manchester decoder logic. The effects of the first factor would be significantly reduced by implementing the video circuits on printed circuit cards. The effects of the second factor could be reduced by using higher absolute accuracy clocks or preferably by improving relative accuracy via the use of phase-lock-loop clock circuits.

The Video Encoder and Decoder were tested for data recovery/reliability at eight encoding frequencies from 1.432 megahertz to 4.00 megahertz. The testing used four hardware configurations. The first configuration was Direct Recovery with the Encoder output routed directly to the Decoder via about 40 feet of coax cable. The second configuration was VTBC Direct Recovery where VTBC processing occurred before routing to the Decoder. The third configuration was VCR Recovery which recorded the Encoder output on a VCR with the VCR video playback routed to the Video Decoder. The fourth configuration was VCR/VTBC Recovery which

processed the VCR playback through the VTBC before routing to the Video Decoder. All four configurations used a Tektronix Model 1470 Video Generator as the video source for the Video Encoder. The VCR used was a Panasonic Model NV-9200 Video Cassette Recoder. The VTBC used was a Microtime Model 2020 Video Time Base Corrector. The VTBC had an effective bandwidth of 4 megahertz. In spite of misleading specifications, the VCR proved to have an effective luminance bandwidth of about 1.8-2.0 megahertz where we had been expecting closer to a 3 megahertz bandwidth. This caused our test plan to become somewhat abbreviated from our original expectations. The test results for data recovery are given in Table 1. The encoding test sample sizes for each case varied from about one million words for the high error rate cases to over 50 million words for the 100% recovery cases. The data reliability of the recovery process was found to be 100% in that no cases of data error were found that had not been identified and flagged as errors by the Video Decoder hardware in a test sample of over 50 million words. The Direct Recovery data testing demonstrated no significant sensitivity to the size or the position of the encoding envelope or to the content of the video image. However for VCR Recovery, the optimum position of the encoding envelope is at the top of the video field. The only detectable error pattern was that about 25% of the errors would occur on the first raster of the encoding envelope with the remaining errors distributed fairly evenly. In our opinion this first-raster problem is primarily due to the horizontal GEN-Lock jitter resulting from relative clock accuracies. Also, the principle of repeat encoding the format line raster as the first and last lines of an encoding envelope was validated. The odds of unrecoverable loss of the format information was approximately one order of magnitude less than the overall error rate for any given recovery case. Overall, the video encoding format selected by us achieved the design goals of supporting the maximum data recovery for any given error condition.

TABLE 1

DATA RECOVERY TEST RESULTS

ENCODING FREQUENCY (MEGAHERTZ)	DIRECT RECOVERY	VTBC DIRECT RECOVERY	VCR RECOVERY	VCR/VTBC RECOVERY
1.432	100%	99%	80%	99%
1.50	100%	99%	30%	70%
1.79	100%	98%	0%	30%
2.00	100%	90%	0%	1%
2.386	100%	50%	*	*
3.00	99.5%	0%	*	*
3.58	85%	0%	*	*
4.00	60%	0%	*	*

*NOT TESTED

SECTION VI

CONCLUSIONS

The BM/VE approach is impractical for recording all activity on a 1553B system. However, in terms of sampled 1553B activity, the BM/VE can incorporate recoverable digital recording capability into a video instrumentation system without affecting the video data recording. This digital recording capability can easily support data acquisition rates of 1200 words per second without loss of image data at sampling rates up to 60 hertz on the older low resolution VCRs (3/4" U-Matic format with 330-line resolution). The sacrifice of only a few lines of video image data can result in large increases in the digital capacity. The primary limit to the BM/VE video encoding capacity is the bandwidth limitations of the recording VCR. Table 2 shows the BM/VE encoding capacity at various encoding frequencies for the case of no image data loss and for the case of loss of normally unused image data. For example, a newer 380-line resolution VCR should easily support a 99% recoverable encoding rate of 1.70 megahertz with a "Case 2" capacity of 3,240 data words per second. Since the BM/VE time-tags data at acquisition, it can be adapted by software to a wide range of instrumentation requirements by trade-off decisions between data sampling rate, data sampling size, encoding frequency, encoding envelope size, and video encoding rate. A change in any of these parameters (within certain limits) can be compensated for by changing one or more other parameters. While we did not have access to VCRs other than Panasonic NV-9200, it is strongly suspected that no off-the-shelf VCR is going to provide greater than 380-line resolution. With the maturing of the "writable-optical-disk" video recording technology, there are reasonable expectations with the next two years for a video recording system that would support BM/VE video encoding at the color subcarrier frequency of 3.58 megahertz. This encoding frequency would allow a "Case 2" capacity of 7,560 data words per second. Regardless of any improvement in VCR bandwidth, we have concluded that it would not be practical to attempt video encoding at any frequency significantly higher than 4 megahertz. While the BM/VE Video Encoder could encode at a 5 megahertz frequency (12 megahertz "phase" clock), it was found that

TABLE 2

BM/VE ENCODING CAPACITY

DATA ENCODING FREQUENCY (MEGAHERTZ)	DECODING CLOCK (MEGAHERTZ)	MAXIMUM WORD CELLS PER RASTER	MAXIMUM CAPACITY CASE 1 (SEE NOTE 1)		MAXIMUM CAPACITY CASE 2 (SEE NOTE 2)	
			RATE (WORDS/SEC)	DATA PERCENTAGE	RATE (WORDS/SEC)	DATA PERCENTAGE
1.432	17.18	3	2160	55.6%	3600	60%
1.50	18.0	3	2160	55.6%	3600	60%
1.79	21.48	4	2760	65.2%	4680	69.2%
2.00	24.00	4	2760	65.2%	4680	69.2%
2.386	28.64	5	3360	71.4%	5760	75%
3.00	36.00	6	3960	75.8%	6840	78.9%
3.58	42.95	8	5160	81.4%	9000	84%
4.00	48.0	9 (NOTE 3)	5760	83.3%	10100	85.7%

NOTE 1: Assuming no loss of image data. Encoding envelope contains 12 rasters (with 16 format words) with position at top of video field.

NOTE 2: Assuming use of off-screen video. Encoding envelope contains 20 rasters (with 24 format words) with position at top of video field.

NOTE 3: The actual hardware limit was 8 words per raster.

even video coax cable (10 feet) would attenuate/distort the encoding envelope beyond any reasonable hope of recovery. Other problems that become very serious above 4 megahertz are video amplifier bandwidths/slew-rates, RF noise coupling, and attempting to implement a manchester decoder that would use a 72 megahertz decoding clock (for 6 megahertz data).

Several specific conclusions on the BM/VE approach can also be made from our experience with the laboratory prototype. The use of phase-lock-loop reference clocks to improve relative clock accuracy between the video source, BM/VE, and the Video Decoder should be considered a requirement with the video encoding frequency being a higher harmonic of the horizontal raster. For data recovery from a VCR recording, the use of a video time base corrector should be considered mandatory. It was also found that the data recovery error rate was very sensitive to the condition of the magnetic tape used on the VCR. In other words, always use top-quality relatively new tape cassettes. The equations given below provide an experimental approximation for determining the maximum video encoding frequency and the maximum number of encodable word cells on a horizontal raster for any given bandwidth of a VCR.

<u>where</u>: F_E = Encoding Frequency (data bit rate)
R_H = VCR Horizontal Resolution (lines/raster)
F_R = Horizontal Raster Frequency (15.75 KHZ)
C_{RW} = Count of Raster Words (Integer)

<u>then</u>:

$$F_E = \frac{R_H F_R}{3.5}$$

$$C_{RW} = \frac{50F_E}{22{,}000{,}000}$$

If C_{RW} is less than 3, the VCR cannot be used with the BM/VE encoding format. The performance of the Video-Encoder and the Video Decoder in the Direct Recovery configuration also suggests a non-BM/VE application of the video encoding technique. In a simulation environment where computer controlled graphics are used to send a video signal to the actual simulator, the video encoding technique could be used to transfer digital simulation data via the video signal to the simulator.

In conclusion, the BM/VE project has determined the performance parameters and design requirements necessary to develop a flyable BM/VE prototype. We estimate that such a flyable prototype and a minimal data recovery system (ground support) could be constructed from off-the-shelf components in 12 to 16 months with the investment of $50,000 and 4 manyears of effort. The resulting prototype would have a volume of less than one cubic foot and would require about 300 Watts of power. The prototype would provide a digital instrumentation capacity to a CTVS instrumentation system of at least 1200 data words per second. Depending on the application, the instrumentation capacity might easily be in the range of 2000 to 3000 words per second. A prototype BM/VE would be particularly applicable to flight testing of fire-control systems.

☆U.S. GOVERNMENT PRINTING OFFICE—759-062-716